5-2003

Test Suite for Multiobjective Optimization and Results Using Normal Boundary Intersection (NBI) in Design Explorer

Siva Kumar Natarajan
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/gradreports

Part of the Mathematics Commons

Recommended Citation
Natarajan, Siva Kumar, "Test Suite for Multiobjective Optimization and Results Using Normal Boundary Intersection (NBI) in Design Explorer" (2003). All Graduate Plan B and other Reports. 1300.
https://digitalcommons.usu.edu/gradreports/1300
TEST SUITE FOR MULTIOBJECTIVE OPTIMIZATION AND RESULTS USING
NORMAL BOUNDARY INTERSECTION (NBI) IN DESIGN EXPLORER

by

Siva Kumar Natarajan

A report submitted in partial fulfillment
of the requirements for the degree
of
MASTER OF SCIENCE
in
Industrial Mathematics

UTAH STATE UNIVERSITY
Logan, Utah
2003
Several methods have been developed to solve multiobjective optimization problems (MOP’s). One of these, Normal Boundary Intersection (NBI), is a method developed by John Dennis and Indraneel Das. NBI is used at The Boeing Company as a tool to solve MOP’s. This report presents a test suite of MOP’s that I developed for Boeing during my internship in summer 2003.

The problems in the test suite were chosen to represent the different types of multiobjective optimization problems that could arise in practice and the complexities involved in solving them. These problems range from those that have nice convex Pareto surfaces to those that have complex disconnected Pareto surfaces. We study whether or not NBI can solve these problems, and the difficulties that arise. The suite also includes real world examples, in particular a truss design problem.

(44 pages)
DEDICATION

I dedicate this report to my mother Ms. Vijayalakshmi Natarajan.
ACKNOWLEDGMENTS

I thank my major professor Dr. Kathryn Turner for her constant guidance and support throughout my master’s program. I thank Dr. Emily Stone for helping me get the opportunity to work with Boeing. I would also like to thank Dr. Vicki Allan for her useful insights on the content of this report. I thank my mentors at The Boeing Company - Joerg Gablonsky, Evin Cramer and John Dennis for their guidance. This project was supported by the NSF grant (DMS 0104818), Department of Mathematics and Statistics, Utah State University and The Boeing Company.

Siva Kumar Natarajan
CONTENTS

ABSTRACT .................................................. iii
DEDICATION ............................................... iv
ACKNOWLEDGMENTS ...................................... v
LIST OF FIGURES .......................................... viii

1 INTRODUCTION ........................................... 1

2 MULTIOBJECTIVE OPTIMIZATION ......................... 2
   2.1 Problem Formulation ................................ 2
   2.2 Some Traditional Methods to Solve MOPs ............ 3
   2.3 Measures of Performance ............................ 4

3 NORMAL BOUNDARY INTERSECTION ...................... 6
   3.1 Some Terminology .................................. 6
   3.2 Example ........................................... 8
   3.3 The Basic Concept of NBI and Problem Formulation . 8
   3.4 Performance features of NBI ....................... 10

4 DESIGN EXPLORER ....................................... 12
   4.1 Tools in Design Explorer ............................ 12
      4.1.1 SPOTS ........................................ 12
      4.1.2 DACEPAC ..................................... 13
      4.1.3 SEQOPT ....................................... 13
      4.1.4 NBI ........................................... 14

5 TEST SUITE ............................................. 16
   5.1 Types of Problems in the Suite ..................... 16
   5.2 Test Problems ....................................... 18
      5.2.1 MOP 1 ......................................... 18
      5.2.2 MOP 2 ......................................... 20
      5.2.3 MOP 3 ......................................... 21
      5.2.4 MOP 4 ......................................... 23
      5.2.5 MOP 5 ......................................... 25
      5.2.6 MOP 6 ......................................... 27
      5.2.7 MOP 7 ......................................... 30
      5.2.8 MOP 8 ......................................... 32

6 CONCLUSIONS AND FUTURE WORK ....................... 36
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Example of a bi-loss map.</td>
<td>9</td>
</tr>
<tr>
<td>3.2</td>
<td>Example of a point obtained by NBI that is not Pareto optimal.</td>
<td>10</td>
</tr>
<tr>
<td>4.1</td>
<td>The Optimization Algorithm</td>
<td>15</td>
</tr>
<tr>
<td>5.1</td>
<td>Pareto plot for MOP 1</td>
<td>18</td>
</tr>
<tr>
<td>5.2</td>
<td>Bi-loss map for MOP 1</td>
<td>19</td>
</tr>
<tr>
<td>5.3</td>
<td>The feasible region for MOP 1</td>
<td>19</td>
</tr>
<tr>
<td>5.4</td>
<td>All Pareto points for MOP 1</td>
<td>20</td>
</tr>
<tr>
<td>5.5</td>
<td>Pareto plot for MOP 2</td>
<td>21</td>
</tr>
<tr>
<td>5.6</td>
<td>Bi-loss map and feasible region for MOP 2</td>
<td>21</td>
</tr>
<tr>
<td>5.7</td>
<td>Pareto plot for MOP 3</td>
<td>22</td>
</tr>
<tr>
<td>5.8</td>
<td>Bi-loss map for MOP 3</td>
<td>23</td>
</tr>
<tr>
<td>5.9</td>
<td>Pareto plot for MOP 4</td>
<td>24</td>
</tr>
<tr>
<td>5.10</td>
<td>Bi-loss map for MOP 4</td>
<td>24</td>
</tr>
<tr>
<td>5.11</td>
<td>The feasible region for MOP 4</td>
<td>25</td>
</tr>
<tr>
<td>5.12</td>
<td>Pareto plot for MOP 5</td>
<td>26</td>
</tr>
<tr>
<td>5.13</td>
<td>Bi-loss map for MOP 5</td>
<td>27</td>
</tr>
<tr>
<td>5.14</td>
<td>A 3-bar truss.</td>
<td>28</td>
</tr>
<tr>
<td>5.15</td>
<td>Pareto plot for MOP 6</td>
<td>29</td>
</tr>
<tr>
<td>5.16</td>
<td>Pareto plot for MOP 7</td>
<td>31</td>
</tr>
<tr>
<td>5.17</td>
<td>Bi-loss map for MOP 7</td>
<td>32</td>
</tr>
<tr>
<td>5.18</td>
<td>Vibrating platform apparatus</td>
<td>33</td>
</tr>
<tr>
<td>5.19</td>
<td>Pareto plot for MOP 8</td>
<td>34</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

Several methods have been developed to solve multiobjective optimization problems, each of which has advantages and drawbacks. It is useful to document some typical example problems that a method can solve, where the method could fail to obtain a solution, or modifications that could be made to the algorithm to overcome a particular problem. This report presents one such suite of problems, which are solved using Normal Boundary Intersection (NBI), a method developed by Indraneel Das and J. E. Dennis Jr [8] to solve multiobjective optimization problems. These problems are chosen so as to represent the different types of multiobjective optimization problems that could arise in practice and the complexities involved in solving them. The suite also includes real world examples.

Design Explorer is a suite of tools developed by researchers at Boeing Mathematics & Computing Technology (M&CT) and Rice University for optimization of approximate models. NBI is one of the tools used in Design Explorer to solve multiobjective optimization problems. The problems that are presented in this report have been solved using NBI in Design Explorer, and the performance of NBI, and the complexities involved in the problems have been outlined. This may give the user more confidence in using Design Explorer, especially if he or she has a multiobjective problem similar to one of the problems in the test suite.

An introduction to multiobjective optimization, some traditional methods that are used to solve them, and measures of performance for multiobjective optimization are discussed in Chapter 2. Chapter 3 describes NBI and compares its performance to that of other methods. Chapter 4 is a brief overview of Design Explorer and its tools. The last part of the report is the test suite. This section presents multiobjective optimization problems including the problem formulation with objectives and constraints, bounds on decision variables and results obtained in Design Explorer.
CHAPTER 2
MULTIOBJECTIVE OPTIMIZATION

Techniques for mathematical optimization are employed in various aspects of design and decision making in fields ranging from mechanical and chemical engineering to finance and political science. As these techniques are applied to practical problems, it is often observed that the final design is guided by more than one objective or criteria. Consider the example of designing a bridge [8]. The structural engineer wants to minimize the total mass of the bridge but also realizes that he has to design it to have maximum stiffness. Since it is generally not possible to obtain a design that would optimize both of these objectives at the same time, the designer needs to assess trade-offs in choosing the final design. Multiobjective optimization quantifies the trade-offs among competing objectives to assist in making design decisions.

We may view an attribute of the design as either a requirement to be imposed by a constraint or an entity to be optimized subject to restrictions. We can even use these methods for robust design where one of the objectives captures a measure of robustness. Multiobjective optimization algorithms formulate a problem as a sequence of problems that can be solved by a single objective algorithm.

2.1 Problem Formulation

A multiobjective optimization problem can be defined as [8]:

$$\min_{x \in C} F(x) = \left[ \begin{array}{c} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{array} \right], \quad n \geq 2 \quad (MOP)$$

$$C = \{ x : h(x) = 0, g(x) \leq 0, a \leq x \leq b \}$$ and $$a \in (\mathbb{R} \cup \{-\infty\})^N, b \in (\mathbb{R} \cup \{\infty\})^N, F : \mathbb{R}^N \mapsto \mathbb{R}^n, h : \mathbb{R}^N \mapsto \mathbb{R}^{n_e}, g : \mathbb{R}^N \mapsto \mathbb{R}^{n_i}$$ are continuously differentiable mappings, N is the number of variables, n is the number of objectives, ne is the number of equality constraints and ni is the number of inequality constraints.

Without loss of generality, we will always refer to the MOP as a minimization problem unless stated otherwise. Usually, in MOP the objectives are at least partly conflicting with
one another. (Otherwise we can solve separate single objective optimization problems taking one objective at a time.) Since no single \( x^* \) would generally minimize every single objective simultaneously, we will use the concept of Pareto optimality. We first define dominance, then define Pareto optimality.

A vector \( F(x) \) is said to dominate another vector \( F(x) \), denoted by \( F(x) < F(x) \) if and only if \( f_i(x) \leq f_i(x) \) for all \( i \in \{1, 2, \ldots, n\} \) and \( f_j(x) < f_j(x) \) for at least one \( j \in \{1, 2, \ldots, n\} \). A point \( x^* \in C \) is said to be globally Pareto optimal or a globally efficient point for MOP if and only if there does not exist \( x \in C \) satisfying \( F(x) < F(x) \).

Thus a design point is a Pareto point if it satisfies all the constraints and additional changes to the design variables cannot improve all of the objectives. The Pareto surface is the set of all Pareto points. Only points on the Pareto surface can be considered reasonable optimizers of a multi-objective problem because any improvement in one objective takes place only if at least one other objective worsens.

2.2 Some Traditional Methods to Solve MOPs

Several approaches have been studied to solve MOPs. These methods can be classified into two categories. The first class of methods involves forming a single objective function, subject to constraints. Once the objective function that reflects a priori preferences of the designer is formulated, the problem is solved as a standard constrained optimization problem. The goal programming method [15] and physical programming [13] are examples of such methods. The main drawback in these methods is the formulation of the objective function.

A family of single objective constrained optimization problems may be generated by forming the objective functions as weighted sums of the multiple objectives. The designer then selects a design from among the solutions to the single objective problems. Typically one would like to vary the weights such that each combination of weights results in a Pareto solution. The designer would like to see Pareto points evenly spread on all parts of the Pareto surface. Hence, one would hope to produce an even spread of points on the Pareto surface using an even distribution of weights. The popular weighted sums method [12, 11]
suffers from the drawback of not producing an even spread of Pareto solutions. Another drawback of the method is that it fails to find points on the nonconvex part of the Pareto surface. Normal Boundary Intersection (NBI), a method developed by Indraneel Das and J. E. Dennis Jr [1], is successful in producing an evenly distributed set of Pareto points given an evenly distributed set of weights. It has been proven [1] that this method produces Pareto points in the nonconvex parts of the Pareto curve and is independent of the relative scales of the objective functions.

2.3 Measures of Performance

Ideally a method should be efficient and should be applicable to a wide class of problems. This section discusses desirable attributes of a multiobjective optimization method. For the purpose of this report we will refer to a Pareto point as the vector of function values of the objectives, and not to the independent variable or the design.

The optimization algorithm should be:

- Capable of producing an even spread of Pareto points. This attribute is important as it provides the assurance that the Pareto points found represent all regions of the efficient part of the design space. This allows the designer to explore the entire efficient design space and to select the appropriate design.

- Capable of capturing any concave regions present in the Pareto curve. It is useful to find points on this region since it constructs a smoother approximation to the boundary of the bi-loss map containing the Pareto points. It is known that the weighted sums method does not produce Pareto points in nonconvex parts of the Pareto curve [4]. Hence the method fails to capture efficient points in an entire portion of the Pareto curve.

- Able to capture non-connected Pareto fronts. Non-connectivity includes non-connectivity of the feasible region in the design space or/and the function space.

- Independent of the relative scales of the objective functions. For a badly scaled
problem [10], the Weighted sums method fails to represent an entire portion of the Pareto curve.

- Capable of producing only Pareto points.
- Easy to use.
- Rapidly convergent to the solution.
CHAPTER 3
NORMAL BOUNDARY INTERSECTION

The Normal Boundary Intersection (NBI) algorithm was introduced by Indraneel Das and J. E. Dennis Jr [8]. NBI is a method for finding several Pareto optimal points for a general multiobjective optimization problem with two or more objectives. Such points collectively capture the trade-off among various conflicting objectives. It has been proven [8] that this method is independent of the relative scales of the objective functions and is successful in producing an evenly distributed set of points in the Pareto set given an evenly distributed set of parameters.

3.1 Some Terminology

Definition: Shadow Minimum: The shadow minimum or utopia point $F^*$, is defined as the vector containing the individual global minima, $f_i^*$, of the objectives within the feasible region.

$$F^* = \begin{bmatrix} f_1^* \\ f_2^* \\ \vdots \\ f_n^* \end{bmatrix}$$  \hspace{1cm} (3.1)

Hence we assume the existence of a minimizer for each of our objectives. The shadow minimum is the solution to a MOP in the rare case when a single $x$ minimizes all the objective functions at the same time. In practical situations, we hope to get as "close" as possible to the shadow minimum, and illuminate the trade-off among the multiple objectives.

Definition: Convex Hull of Individual Minima (CHIM) [8]: Let $x_i^*$ be the respective global minimizers of $f_i(x)$, $i = 1, 2, \ldots, n$ over $x \in C$. Let $F_i^* = F(x_i^*)$, $i = 1, 2, \ldots, n$. Let $\Phi$ be the $n \times n$ matrix whose $i^{th}$ column is $F_i^* - F^*$, known as the pay-off matrix. Then the set of points in $\mathbb{R}^n$ that are convex combinations of the columns of $\Phi$, i.e., $\{\Phi \beta : \beta \in \mathbb{R}^n, \sum_{i=1}^n \beta_i = 1, \beta_i \geq 0\}$, is referred to as the Convex Hull of Individual Minima.

Definition: Objective Space: The set of attainable objective vectors $\{F(x) : x \in C\}$ is denoted by $\Gamma$. $C$ is mapped by $F$ onto $\Gamma$. The space $\mathbb{R}^n$, which contains $\Gamma$ is referred to
as the objective space.

**Definition:** Multi-loss map [8]: The map of $C$ under $F$ in the objective space is called the multi-loss map (bi-loss map, if $n = 2$). We shall denote the boundary of $\Gamma$ by $\partial \Gamma$. The set of all Pareto optimal points is denoted by $\Psi$. The complete curve/surface of Pareto minima (continuous or not) is defined as the *trade-off function*.

**Definition:** $CHIM_+$: Let $CHIM_{\infty}$ be the affine subspace of lowest dimension that contains the CHIM, i.e., the set $\{\Phi \beta : \beta \in \mathbb{R}^n, \sum_{i=1}^{n} \beta_i = 1\}$. The $CHIM_+$ is defined as the convex hull of the points in the intersection of $\Gamma$ and $CHIM_{\infty}$. In other words, consider extending (or withdrawing) the boundary of the CHIM simplex to touch $\partial \Gamma$. The ‘extension’ of CHIM thus obtained is defined as $CHIM_+$.

**Definition:** Barycentric coordinates[1]:
Let $v_1, v_2, \ldots, v_k$ be any basis for a vector space $\subset \mathbb{R}^n$, let $O$ be a point in $\mathbb{R}^n$.
Define the affine space $A$ as
\[
\{ x \in \mathbb{R}^n | x = O + c_1 v_1 + c_2 v_2 + \ldots + c_k v_k \}
\]
for some scalars $c_1, c_2, \ldots, c_k$.

The representation of each point in $A$ in terms of $O$ and the vectors $v_1, v_2, \ldots, v_k$ is unique.
Define points $P_i$ by
\[
\begin{align*}
P_0 &= O \\
P_1 &= O + v_1 \\
P_2 &= O + v_2 \\
&\vdots \\
P_k &= O + v_k
\end{align*}
\]
and suppose a point $P \in A$ has the representation
\[
p = O + p_1 v_1 + p_2 v_2 + \ldots + p_k v_k
\]
Let
\[ p_0 = 1 - (p_1 + p_2 + \ldots + p_k). \]

Then \( P \) can be equivalently written as
\[ P = p_0 P_0 + p_1 P_1 + p_2 P_2 + \ldots + p_k P_k \]
where
\[ p_0 + p_1 + p_2 + \ldots + p_k = 1 \]

In this form, the values \((p_0, p_1, p_2, \ldots, p_k)\) are called the barycentric coordinates of \( P \) relative to the points \((P_0, P_1, P_2, \ldots, P_k)\).

3.2 Example

Consider the example of a bi-objective problem. Figure 3.1 shows an example of a bi-loss map. It is assumed that the objective functions have been defined with the shadow minimum shifted to the origin, so that all the objective functions are non-negative, i.e., \( F(x) \) is redefined as \( F(x) \leftarrow F(x) - F^* \).

Figure 3.1 shows the set \( \Gamma \) in the objective space. The point \( A \) is \( F_1^* \), \( B \) is \( F_2^* \), \( O \) is the shadow minimum (and the origin), the broken line segment \( AB \) is the CHIM, while the arc \( ACB \) is the set of all Pareto minima in the objective space, i.e., it is the trade-off curve. In any bi-objective problem \((n = 2)\), \( \text{CHIM}=\text{CHIM}_+ \). For \( n > 2 \) \( \text{CHIM} \) may not equal \( \text{CHIM}_+ \). For example [8], suppose \( \Gamma \) is a sphere in \( \mathbb{R}^3 \) touching the coordinate axes. Then \( \text{CHIM} \) is the triangle formed by joining the three points where the sphere touches the axes. On the other hand, \( \text{CHIM}_+ \) is the region of intersection of the plane containing the triangle and the sphere, i.e., a circular disc. Hence \( \text{CHIM} \neq \text{CHIM}_+ \).

3.3 The Basic Concept of NBI and Problem Formulation

The goal of NBI is to find the portion of \( \partial \Gamma \) which contains the Pareto optimal points. Let us assume that the shadow minimum \( F^* \) has already been calculated. NBI is based on the idea that the intersection between the boundary \( \partial \Gamma \) and the normal pointing towards
the origin starting from any point on the CHIM is a point of $\partial \Gamma$ containing the efficient points.

In the following, we formulate an optimization problem whose solution will give us points on $\partial \Gamma$. A point on the CHIM is given by $\beta \Phi$ for a particular value of $\beta$. Define barycentric coordinates $\beta, \Phi \beta$ that represent a point on the CHIM. Let $\hat{n}$ denote the unit normal to the CHIM simplex pointing towards the origin. Then $\Phi \beta + t \hat{n}, t \in \mathbb{R}$ represents the set of points on that normal. The point of intersection of the normal and the boundary of $F$ closest to the origin is the global solution to the following subproblem:

$$\begin{align*}
\max_{x,t,t} & \quad t \\
\text{s.t.} & \quad \Phi \beta + t \hat{n} = F(x) \\
& \quad h(x) = 0 \\
& \quad g(x) \leq 0 \\
& \quad a \leq x \leq b
\end{align*}$$

$\text{(NBI}_\beta\text{)}$

The first vector constraint ensures that the point is mapped by $F$ to a point on the normal, while the remaining constraints ensure feasibility of $x$ with respect to the original problem (MOP). Remember that we assumed the shadow minimum $F^*$ to be shifted to the
Figure 3.2. Example of a point obtained by NBI that is not Pareto optimal.

As we can see in the above subproblem, \( \beta \) is the characterizing parameter. As we vary \( \beta \), we find points on the boundary of \( \Gamma \), thus constructing a pointwise approximation of the portion of \( \partial \Gamma \) that contains the Pareto surface.

3.4 Performance features of NBI

The present section compares the general performance of NBI and the Weighted sums method in solving multiobjective problems.

- The points obtained by NBI in solving \((NBI_\beta)\) for various settings of \( \beta \) are Pareto optimal points unless they lie on a sufficiently concave part of the curve as shown in figure 3.2. If the Pareto surface is convex, then the points obtained by NBI are always guaranteed to be Pareto optimal. Most practical multiobjective problems possess this feature. But if points in the concave part of the Pareto surface are Pareto optimal, then NBI finds those points, which is a merit over minimizing convex combinations of objectives [10].
Note that even if some of the points on the portion of \( \partial \Gamma \) that contains the Pareto surface are not Pareto optimal, it is useful to find these points to construct a smoother approximation to this boundary (curve \( ABCDE \) in figure 3.2) of the bi-loss map.

- It is desirable to generate an even spread of Pareto points, representative of all parts of the Pareto set and not clusters of points in certain parts which fail to provide a good idea of the entire shape. In NBI, we select settings of the parameter \( \beta \) such that the points \( \Phi \beta \) form a uniformly spaced grid on the CHIM. The details of selecting such \( \beta \) are described elaborately in [8]. Since NBI points are restricted to lie on a set of parallel normals starting from these uniformly spread points, the projections of the NBI points on the CHIM are uniformly spread. Thus, NBI can yield a good approximation of the Pareto surface. On the other hand, the Weighted Sums method fails to obtain an even spread of Pareto points for an even distribution of weights [10]. An even distribution of weights can result in points that are clustered in space. Hence, caution should be exercised by designers who hope to minimize just one weighted sum of objectives and get a point in the middle region of the Pareto set. NBI should be regarded as a tool for generating points on the Pareto surface which give a better approximation to the overall surface than that obtained by weighted sums.

- It has been proven in [8] that NBI is independent of the relative scales of the objective functions.
CHAPTER 4
DESIGN EXPLORER

Design Explorer is a suite of modeling and optimization tools used to perform design studies and optimization for problems involving expensive computer simulations. A CFD (Computational Fluid Dynamics) code is an example where evaluating the function is very expensive. Hence, it is impractical to perform optimization on the function where it is required to evaluate the function many times. Hence, it is important to have a systematic and efficient method for exploring the design space. Design Explorer is the focus of a multi-year collaboration between researchers at Boeing Mathematics & Computing Technology (M&CT) and Rice University on the topic of optimization of approximate models.

The tools in Design Explorer could be used to:

- Gain insight on the design variables and understand the effect of their variability on the product performance.
- Find optimal designs.
- Perform multiobjective optimization of competing objectives and select the best compromise.

4.1 Tools in Design Explorer

A brief description of selected tools available in Design Explorer[2] follows. The user can make use of the Design Explorer scripts which tie together several of the modeling and optimization tools that are described below. The scripts simplify the process of working with the Design Explorer tools to design an experiment, build a response surface model, gain insight from a model, and optimize a user-defined problem based on the model. The tools in Design Explorer can also be used on their own.

4.1.1 SPOTS

SPOTS[2] is a software tool that is used to perform parameter studies and to parallelize programs (distribute a program to multiple machines) in Design explorer. Parameter studies
or parallelizing involve running a program whose inputs are in the form of ASCII files many times, by modifying the input for each run. This requires rewriting certain “Spots” in the ASCII file. Generally users do this by writing driver programs like sed or awk scripts or C or Fortran programs. Spots makes it very easy to modify these files automatically.

4.1.2 DACEPAC

Design of Experiments (DoE) is a method to perform design studies with complex computer simulations. DACE (Design and Analysis of Computer Experiments [6]) is a particular approach to DoE which is used in design studies, optimization and probabilistic analysis to understand the effect of variations in design and manufacturing on product performance. Design and Analysis of Computer Experiments Package (DACEPAC[2]) allows the user to chose experiment sites, fit models, analyze models, refine models, and perform orthogonal array sampling with distributions.

4.1.3 SEQOPT

Sequential Optimization (SEQOPT[2]) is a global method to perform optimization for functions that are too expensive to optimize directly [4]. It is a combination of modeling tools, model optimization tools, and a controller. The basic approach used by SEQOPT is iterative. A modified flow chart of the method is given in figure 4.1.

The algorithm starts with an initial set of points obtained from a DACE experiment. The true function is evaluated at these points and Kriging models are built for each objective and/or constraint. Then at each SEQOPT iteration, the method identifies points at which true simulation data is needed. The data are obtained and surrogate models are updated using the new data. The optimization controller uses the new data to decide how the optimization process will proceed.

The search mode starts with finding “global” optimizers and points for model improvement. The “global” optimizers are found by running a local optimization code on the model problem from several start points. The objective and constraint functions of the model problem are computed using surrogate models of the computer simulation outputs. Duplicate local optimizers for the model problem are culled out of the set.
Next, a set of points for model-refinement is determined. These CBLGS (Constrained, Balanced, Local-Global Search) points have high modeling errors or have a good chance of being feasible, and a reasonable probability of improving on the best point found. The computer simulation is run at the distinct local optimizers and model-refinement points. The models are then updated to include information from the new runs. If SEQOPT fails to make progress for several iterations, it initiates a poll procedure. During a poll, the computer simulation is run on a subgrid of points surrounding the current best point. SEQOPT stops when its iteration limit is reached or when a poll at the finest grid level fails to improve on the best point.

4.1.4 NBI

Normal Boundary Intersection (NBI) is a method to generate Pareto points for multiobjective problems. The user provides a set of scalar valued objective functions, a set of bounds, and linear or nonlinear constraints. The software constructs a series of new objective functions and constraint sets and solves a sequence of optimization problems. The output is a set of Pareto points in design space and the corresponding objective and constraint values. The user can then decide which of the points best suits his needs. This implementation of NBI in Design Explorer has additional options such as computing extrema for individual objectives and then solving the NBI subproblems. A detailed description of NBI is given in Chapter 3.
Define a domain

Define an experiment

Build a model

Search or Poll

SEARCH
"Global" optimization of models

Determine model improvement Points (CBLGS)

Evaluate the true simulation code at these points

Update Models with new data

POLL

CONTROLLER:
Filter-based Derivative free Optimization Algorithm

Update Iterate count

Update size of Poll grid

Figure 4.1. The Optimization Algorithm.
CHAPTER 5
TEST SUITE

A standard suite of MOPs exhibiting relevant MOP domain characteristics can provide a necessary basis to compare various algorithms. This document describes one such test suite, which consists of problems that exhibit various characteristics of MOPs that real world problems possess. The reader can use this suite to analyze the performance of Design Explorer (NBI) in solving MOPs and compare it with other software packages. Test suites usually contain a number of functions, many of whose origins and rationale are unknown. Hence, documentation of MOP test suites is important and is an asset to MOP research. The following section gives a description of the structure of the test suite and the types of problems that are included in it.

5.1 Types of Problems in the Suite

Problems should only be included in a test suite because they possess typical characteristics of MOPs and not because other researchers have used them. This test suite contains problems that exhibit a variety of properties based on the shape of the Pareto surface, local versus global minimum, scaling and so on. One would expect software to solve certain kinds of problems better than the others. The test suite has been structured such that the reader can get a clear idea of the category of problems that Design Explorer (NBI) has been able to solve.

An algorithm that solves all problems presented in a test suite has no guarantee of continued effectiveness and efficiency when applied to real world problems. This suite contains a few examples of physical MOPs that Design Explorer (NBI) has solved successfully. Two truss design problems and a problem dealing with the design of layers in a vibrating platform (this has serious scaling problems) have been included. These clearly show how real world problems can be hard to solve, compared to made-up analytical problems.

Dimensionality is another property that has to be considered in solving MOPs. Most of the problems in real life involve more than two or three variables. NBI can be applied to
problems with any number of decision variables, perhaps only restricted by considerations of computational expense. The largest problem included in the test suite is the truss design problem, which has eight decision variables.

The solution to an MOP found by a method can also depend on the optimizer used. Software that uses a local optimizer can generate an incorrect Pareto surface compared to one that uses a more global optimizer. It is possible that the software misses an efficient part of the Pareto surface when it finds only a local minimum. Design Explorer (NBI) can be used to find a better local solution for a single objective optimization problem, since it uses a more global optimizer SEQOPT. More details about the optimization method can be found in [4]. An example is included in the test suite that compares Pareto surfaces obtained using OPTLIB (a local optimizer) and SEQOPT (a ‘global’ optimizer).

When implementing NBI, it is (implicitly) assumed that the problem domain has been properly considered and a decision has been made that NBI is an appropriate search algorithm for the given MOP. An MOP domain may consist of continuous or discrete (e.g., integer constrained) functions or even a mix of the two. Here we restrict the discussion to continuous MOPs. The following characteristics have been recognized and the problems are chosen to represent them.

- Domain Space (x-space): Connected or disconnected, scalable.

- Function Space (f-space): Connected or disconnected, convex, concave, or mixed.

In summary, this test suite contains characteristic problems from the algorithm’s problem domain. Some represent real world situations, and others range in difficulty from easy to hard.
5.2 Test Problems

5.2.1 MOP 1

The following problem is taken from [16].

Objectives       $f, g$
Variables        $x, y$
Characteristics  Example of a convex Pareto curve.
Problem          \[
\begin{align*}
\min \quad & f = \frac{1}{x^2 + y^2 + 1} \\
 & g = x^2 + 3y^2 + 1 \\
\end{align*}
\]
Bounds           $-3 \leq x, y \leq 3$

Results and Comments

Convex Pareto curves are typical solutions to MOPs. For this problem, OPTLIB was able to find only a local minimizer $(0, 0)$ when solving for the individual minimum for the function $f$, which also turns out to be the individual minimum for $g$. The function values of $f(0, 0)$ and $g(0, 0)$ are the same and equal to 1.0. Hence the CHIM is only a point and the problem could not be solved. SEQOPT on the other hand found the global minimum for $f$. The CHIM is a line segment and the problem was solved successfully to obtain the Pareto curve shown in figure 5.1.

![Figure 5.1. Pareto plot for MOP 1](image)

The bi-loss map is plotted by evaluating the objectives and constraints over a grid.
of points. The points are plotted only if they are feasible. Figure 5.2 confirms that the points obtained are Pareto points and are evenly spread on the curve. The feasible region (figure 5.3) shows Pareto points in the domain space. The points lined up on the upper bound on the variable $x$ suggests that the solutions are affected primarily by the bounds on the variables.

![Figure 5.2. Bi-loss map for MOP 1](image)

Figure 5.2. Bi-loss map for MOP 1

![Figure 5.3. The feasible region for MOP 1](image)

Figure 5.3. The feasible region for MOP 1

An interesting aspect of this problem is that the solution is non-unique. For every point in the bi-loss map, there are 4 possible solutions (due to symmetry of the objective functions with respect to the $x$ and $y$ axes) as shown by the four symbols in figure 5.4.

In a real-world problem similar to this problem where the solution is non-unique, the
user might have a preference for using one setting of the design variables over another. Hence the user would prefer to have all the possible solutions so that he could choose the best one of them.

5.2.2 MOP 2

The following problem is taken from [16].

| Objectives | \( f, g \) |
| Variables | \( x, y \) |
| Characteristics | Example of a concave Pareto curve. |
| Problem | \[
\begin{align*}
\min & \quad f = 1 - e^{-(x-1)^2-(y+1)^2} \\
& \quad g = 1 - e^{-(x+1)^2-(y-1)^2}
\end{align*}
\] |
| Bounds | \(-4 \leq x, y \leq 4\) |

Results and Comments

The Pareto plot (figure 5.5) compares solutions obtained by two approaches. One approach is to solve the entire problem using OPTLIB. The points thus obtained are denoted by ‘o’. In the second approach, the individual minima for the objectives were found using SEQOPT and the subproblems were solved using OPTLIB. The points obtained are denoted by ‘x’. The plot shows that both approaches do equally well in solving the problem.

The bi-loss map and the feasible region (figure 5.6) is plotted as described in MOP1. The figure confirms that the NBI points obtained are Pareto points. It is to be noted that the shape of the bi-loss map is not due to the constraints, since this problem has only
simple bounds as constraints, but are the result of the behavior of the objective functions with respect to each other.

5.2.3 MOP 3

The following problem is taken from [14].

Objectives \( f, g \)

Variables \( x, y \)

Characteristics Problem with scaling issues.

Problem \( \min \left\{ \begin{array}{l} f = x \\ g = y \end{array} \right\} \)

Constraints \( \left( \frac{x - 20}{20} \right) + (y - 1)^8 \leq 1 \)
Results and Comments

Scaling of objectives, constraints and/or decision variables are important while solving MOPs. This problem is an example that deals with scaling issues where one design metric is orders of magnitude larger than the other. Design Explorer has options to scale the objectives, constraints and decision variables before solving the problem. Since this problem is an illustration of scaling issues, NBI was used on the original (unscaled) problem.

Figure 5.7. Pareto plot for MOP 3

The Pareto plot (figure 5.7) compares solutions obtained by the two approaches using only OPTLIB and SEQOPT with OPTLIB as in MOP2. The results clearly show that NBI is not affected by function scales. It can also be seen that the Pareto points obtained are evenly spread. This is an advantage over the weighted sums method which does not produce an even spread of Pareto points for this problem, but instead produces points that are clustered in the vertical portion of the Pareto curve [10], [14].

Since $f$ and $g$ are equal to $x$ and $y$ respectively, the bi-loss map (figure 5.8) for the problem also shows the feasible region, and confirms that the points obtained are Pareto Points. The two points at the right end of the Pareto curve are an exception as they are dominated points. These points suggest that the optimizer could have found local optima.
5.2.4 MOP 4

The following problem is taken from [17].

Objectives
Variables $f, g$
$x, y$

Characteristics
Problem illustrating local versus global minimum issues.

Problem
\[
\begin{align*}
    f &= (x + y - 7.5)^2 + \frac{(y - x + 3)^2}{4} \\
    g &= \frac{(x - 1)^2}{4} + \frac{(y - 4)^2}{2} \\
    \frac{(x - 2)^3}{2} + y - 2.5 &\leq 0 \\
    x + y - 8(y - x + 0.65)^2 - 3.85 &\leq 0
\end{align*}
\]

Constraints

Bounds

0 \leq x \leq 5
0 \leq y \leq 3

Results and Comments

This problem is a typical example to illustrate problems that could arise when we use a local optimizer. The problem was first solved using the optimizer in MATLAB. It was successful in finding the global minimum $(1, 3)$ for $g$ with a function value of 0.5 but found the local minimum $(2.6390, 2.3695)$ for $f$ with a function value of 8.0713. When the NBI subproblems were solved using the MATLAB implementation of NBI[3], points on only a part of the Pareto curve were obtained.

When OPTLIB was used to solve the same problem, it found the global minimum $(3.01, 1.98)$ for $f$ with a function value of 7.28. But when solving for the individual minimum
for $g$, it found the same point $(3.01, 1.98)$ as above with a function value of 3.05, which is only a local minimum. Hence the CHIM is a just a point and the subproblems could not be solved.

Finally when SEQOPT was used, it resulted in the global minima for both $f$ and $g$. The subproblems were then solved using OPTLIB. As mentioned in Section 3.4, NBI also found dominated points (denoted by ‘.’ in figure 5.9). These dominated points were removed from the solutions using a script called “showNondom” in Design Explorer. Eliminating these points from the solution might not be a preferred thing to do. A designer might want to see the dominated points also as they definitely lie in a region of interest. Hence, this
feature of NBI to obtain dominated points could be considered as an advantage rather than a drawback of the method.

The box shown in the bi-loss map (figure 5.10) is to indicate that the Pareto curve (figure 5.9) shown is a magnified view of the Pareto points in the region of the bi-loss map enclosed within the box. The feasible region is shown in figure 5.11.

As seen in this example, it is very important that we use a more global optimizer like SEQOPT at least to solve for the individual minima of the objective functions. Otherwise there is a chance that the user does not even see a portion of the Pareto curve that contains the efficient points.

5.2.5 MOP 5

The following problem is taken from [16].

Objectives  $f, g$

Variables  $x, y$

Characteristics  Example of a disconnected Pareto curve.

Problem  \[
\begin{align*}
\min \left\{ & f = x \\
& g = y \\
& -x^2 - y^2 + 1 + 0.1 \cos \left[ 16 \arctan \left( \frac{x}{y} \right) \right] \leq 0 \\
& \left( x - \frac{1}{2} \right)^2 + \left( y - \frac{1}{2} \right)^2 \leq \frac{1}{2}
\end{align*}
\]

Bounds  $10^{-6} \leq x, y \leq \pi$
Results and Comments

This example illustrates that NBI can find points on a disconnected Pareto curve. The Pareto plot (figure 5.12 compares solutions obtained using NBI (denoted by 'x'), which includes dominated points and the solution after eliminating the dominated points. As we can see, the Pareto curve (which contains only non-dominated points) is disconnected.

Figure 5.12. Pareto plot for MOP 5

The existence of discontinuities in the Pareto curve is not obvious if we look only at the Pareto points. The discontinuities are definitely regions of interest and the designer might wish to know why Pareto points were not found in those regions. Clearly the dominated points found by NBI (denoted by '*' in the bi-loss map-figure 5.13) give the designer a more
satisfactory insight on the discontinuities. The dominated points make it clear that the bi-loss map is not disconnected. The dominated points lie on the boundary of the bi-loss map and connect the disconnected portions of the Pareto curve.

5.2.6 MOP 6

The following problem is taken from [7].

**Objectives**
- \( d, V \)

**Variables**
- \( \theta, \alpha, x, a_1, a_2, a_3, u_1, u_2 \)

**Characteristics**
- Real-world example with eight decision variables.

**Description**

The problem described here arises in structural optimization. The problem is to find the optimal position of the vertical bar (figure 5.14) of fixed length \( L \) (the bars on the edge get fixed and their lengths decided accordingly) between 1/4 and 3/4 of the entire distance \( D \) and the optimal bar cross-sectional areas. The structure is subjected to a wind load \((W_1)\) and suspended load \((W_2)\). The angles \( \theta \) and \( \alpha \) clearly depend on the chosen location \( x \).

Other optimization variables are the cross-sectional areas of the bars \( a_1, a_2, a_3 \), which are allowed to vary between 0.8 \( \text{in}^2 \) and 3.0 \( \text{in}^2 \). Let \( u_1, u_2 \) denote respectively the horizontal and vertical displacements of the node \( P \); \( d_1, d_2, d_3 \) are the elongations of the three bars respectively, and \( E \) is the modulus of elasticity of the materials of the bars. The first objective that is minimized is the total displacement at node \( P \) denoted by \( d \). The square of the displacement is taken to satisfy differentiability everywhere. The second objective is the total volume of the structure \( V \). Other variables in the problem \( s_1, s_2, s_3 \) denote the stresses in the left, middle and right bars respectively. The details of the derivations of the objectives and constraints given below can be found in [7].

**Problem**

\[
\begin{align*}
\min \left\{ f &= u_1^2 + u_2^2 \\
g &= a_1 \frac{L}{\sin \theta} + a_2 L + a_3 \frac{L}{\sin \alpha}
\right\}
\end{align*}
\]
Figure 5.14. A 3-bar truss.

Constraints

\[
\begin{bmatrix} K_1 & K_2 \\ K_3 & K_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}
\]

\[x = L \cot \theta\]

\[D - x = L \cot \alpha\]

\[|s_i| \leq s_{max}, i = 1, 2, 3\]

More Information

\[K_1 = \left( \frac{E}{L} \right) (a_1 \sin \theta \cos^2 \theta + a_3 \sin \alpha \cos^2 \alpha)\]

\[K_2 = \left( \frac{E}{L} \right) (a_1 \cos \theta \sin^2 \theta - a_3 \cos \alpha \sin^2 \alpha)\]

\[K_3 = \left( \frac{E}{L} \right) (a_1 \cos \theta \sin^2 \theta - a_3 \cos \alpha \sin^2 \alpha)\]

\[K_4 = \left( \frac{E}{L} \right) (a_1 \sin^3 \theta + a_2 + a_3 \sin^3 \alpha)\]

\[E = 4.176 \times 10^6 \text{kips/ft}^2\]

\[L = 60 \text{ft}\]

\[D = 120\]

\[W_1 = 100 \text{kips}\]

\[W_2 = 1000 \text{kips}\]

\[s_{max} = 79,200 \text{kips/ft}^2\]

\[s_1 = \left( \frac{E}{L} \right) |d_1| \sin \theta\]

\[s_2 = \left( \frac{E}{L} \right) |d_2|\]

\[s_3 = \left( \frac{E}{L} \right) |d_3| \sin \alpha\]

\[d_1 = u_1 \cos \theta + u_2 \sin \theta\]

\[d_2 = u_2\]

\[d_3 = -u_1 \cos \alpha + u_2 \sin \alpha\]

Bounds

\[0.005556 \leq a_i \leq 0.020833, i = 1, 2, 3\]

\[0.588003 \leq \theta, \alpha \leq 1.107149\]

\[30 \leq x \leq 90\]

\[-1.2 \leq u_1, u_2 \leq 1.2\]
Results and Comments

The bi-objective problem was solved for 21 evenly spaced points on the CHIM. The last constraints were implemented in the form $s_i \leq s_{\text{max}}$ and $s_i \geq -s_{\text{max}}, i = 1, 2, 3$. Design Explorer (NBI) was successful in solving for the Pareto points. One should keep in mind that this is an 8 dimensional problem and requires more computation time than the other analytical problems that were presented earlier. SEQOPT could not solve for the individual minima, as it could not resolve the feasible region. The feasible region is so small that SEQOPT was not able to find a feasible point. While solving the problem, OPTLIB could not resolve the projected gradient, the tolerance of which was set to a default value of $10^{-5}$. The tolerance had to be changed to $7 \times 10^{-5}$ for the optimization to converge.

![Figure 5.15. Pareto plot for MOP 6](image.png)

The two points shown in the Pareto plot that are in the circle form an unexpected bulge in the Pareto curve. This might point to local optima problems similar to the ones mentioned in MOP 4.

5.2.7 MOP 7

The following problem in taken from [5].

Objectives $f, g$
Variables $x_1, x_2, y$
Characteristics A real-world two-bar Truss problem with serious scaling issues.
Description

This example involves the design of a two-bar truss. This truss is similar to the truss described in MOP 8 without the vertical bar. Here point P, which is subjected to a force of 100kN in the downward direction, is to be located vertically and the cross-sectional areas of the bars on the edge are to be selected. $x_1$ and $x_2$ represent the length of the left and right bars respectively. If the vertical height of P is given by $y$, the design variables are $x_1$, $x_2$ and $y$. The objectives that are to be minimized are the total volume of the truss material and the stress in the left bar. The constraints require that the stresses in the bars on the edge not exceed 100,000 kPa and the total volume of the material not exceed $0.1m^3$. It can be seen from the equations of the objective that, in order to generate Pareto optimal solutions in a reasonable range, objective constraints need to be imposed.

Problem

$$\min \begin{cases} f = x_1\sqrt{16 + y^2} + x_2\sqrt{1 + y^2} \\ g = \frac{20\sqrt{16 + y^2}}{yx_1} \end{cases}$$

Constraints

$$f \leq 0.1$$
$$g \leq 100,000$$
$$\frac{80\sqrt{1 + y^2}}{yx_2} \leq 100,000$$

Bounds

$$0.00082462 \leq x_1 \leq 0.0192844$$
$$0.0011314 \leq x_2 \leq 0.030319$$
$$1.0 \leq y \leq 3.0$$

Results and Comments

In the similar two bar truss problem as given in [5], there were no bounds on $x_1$ and $x_2$ (except that they have to be positive numbers). Since Design Explorer requires bounds specified on the variables, appropriate bounds were imposed. These bounds were derived from the constraints on $f$ and $g$. 
Figure 5.16. Pareto plot for MOP 7

It can be seen from the Pareto plot that this problem has serious scaling issues. \( f \) is of the order of \( 10^{-2} \) and \( g \) is in the order of \( 10^4 \). The variables are also not of the same order of magnitude. Design Explorer has options to scale the objective functions and the variables before solving the problem. To obtain the above solution, the following alterations were made to the problem definition.

- \( f \) was scaled by \( 10^3 \).
- \( g \) was scaled by \( 10^{-3} \).
- \( x_1 \) was scaled by \( 10^3 \).
- \( x_2 \) was scaled by 500.
- Number of function evaluations was increased to 5000 (from the default value of 2000)
- Number of iterations was increased to 300 (from the default value of 75).

The Pareto plot (figure 5.16) shows the trade-off between the two objectives. It is observed that both approaches described earlier (using only OPTLIB and OPTLIB with SEQOPT) do equally well in solving the problem. The bi-loss map (figure 5.17) is plotted as described in MOP1. The figure confirms that the NBI points obtained are Pareto points.
5.2.8 MOP 8

The following problem in taken from [5].

Objectives \( f, g \)

Variables \( d_1, d_2, d_3, b, L \)

Characteristics Vibrating platform problem.

Description

The problem is to design a platform (figure 5.18) with a motor mounted on it. The platform is made of a combination of three materials (denoted by material 1, 2 and 3), material 3 being the surface of the platform on which the motor is mounted. The machine setup is simplified as a pin-pin supported beam carrying a weight. A vibratory disturbance is imparted from the motor onto the beam, which is of length \( L \), width \( b \) and symmetrical about its mid-plane. Variables \( d_1 \) and \( d_2 \) represent the distance of the midplane from the contact of materials 1 and 2, and 2 and 3 respectively. Variable \( d_3 \) represents the distance of the midplane from the top of the beam. \( \rho \) is the mass density, \( E \) is the Young’s modulus of elasticity and \( c \) is the cost per unit volume of the materials, the values of which are given below. The objective is to design a sandwich beam in order to minimize the vibration (maximize the fundamental frequency) of the beam ‘\( f \)’ that results from the motor disturbance and the cost of the setup ‘\( g \)’.
Figure 5.18. Vibrating platform apparatus.

**Problem**

\[
\begin{align*}
\text{min} & \quad f = -\left(\frac{\pi}{2L^2}\right) \left(\frac{EI}{\mu}\right)^{1/2} \\
& \text{subject to} \\
& g = 2b [c_1 d_1 + c_2 (d_2 - d_1) + c_3 (d_3 - d_2)]
\end{align*}
\]

**Constraints**

\[
\begin{align*}
\mu L - 2800 & \leq 0 \\
d_2 - d_1 & \leq 0.01 \\
d_3 - d_2 & \leq 0.01
\end{align*}
\]

**More Information**

\[
\begin{align*}
EI &= (2b/3) [E_1 d_1^3 + E_2 (d_2^3 - d_1^3) + E_3 (d_3^3 - d_2^3)] \\
\mu &= 2b [\rho_1 d_1 + \rho_2 (d_2 - d_1) + \rho_3 (d_3 - d_2)] \\
\rho_1 &= 100 \text{kg/m}^3, \rho_2 = 2770 \text{kg/m}^3, \rho_3 = 7780 \text{kg/m}^3 \\
E_1 &= 1.6 \times 10^9 \text{N/m}^2, E_2 = 70 \times 10^9 \text{N/m}^2, E_3 = 200 \times 10^9 \text{N/m}^2 \\
c_1 &= $500/\text{m}^3, c_2 = $1500/\text{m}^3, c_3 = $800/\text{m}^3
\end{align*}
\]

**Bounds**

\[
\begin{align*}
0.05 & \leq d_1 \leq 0.5 \\
0.05 & \leq d_2 \leq 0.2 \\
0.2 & \leq d_3 \leq 0.6 \\
0.35 & \leq b \leq 0.5 \\
3 & \leq L \leq 6
\end{align*}
\]
Results and Comments

The problem can be reformulated by changing the order of the layers in the sandwich beam (there can be 6 such combinations). This is a particular case of the beam with the above arrangement of the layers.

![Figure 5.19. Pareto plot for MOP 8](image)

The objectives that are minimized are the negative of the fundamental frequency and the cost per volume of the setup. The Pareto plot shows the trade-off between the two objectives. While solving for the individual minima for the objectives, OPTLIB found a local minimum for \( f (-4.26986082e+02) \) and \( g (7.00000000e+01) \). SEQOPT on the other hand found the global minimum for \( f (-4.30978552e+02) \) and \( g (6.97949218e+01) \). Hence we observe the difference in the Pareto points generated by the two methods (described in MOP 2).

If a designer were to select an optimal design given the Pareto points obtained, he/she would be most likely to select one which is in the middle region of the Pareto curve. A point of maximum bulge in this region is referred to as the knee of the Pareto curve [9]. It can be seen that even though OPTLIB found only the local minima, it still produced points in the region of interest.
CHAPTER 6
CONCLUSIONS AND FUTURE WORK

A test suite has been presented to represent a sample of multiobjective optimization problems that arise in practice. These problems, including some real-world examples, have been solved using NBI from Design Explorer environment. NBI can solve these problems satisfactorily. It produces Pareto points that are evenly spread and is independent of the relative scales of the objective functions. Furthermore, NBI can be used to solve MOPs with any number of objectives and variables, perhaps only restricted by computational expense.

NBI produces dominated points for some problems that have nonconvex Pareto surfaces. Post-processing of results to identify dominated points (if any) that are produced by NBI can also be done. Sometimes the optimizer had serious problems with local optima. Hence it is important that a more global optimizer like SEQOPT be used to solve for the individual minima of the objective functions, to ensure that the user does not miss a major portion of the Pareto curve. It might also be beneficial to use a more global optimizer to solve the NBI subproblems.
REFERENCES


