ABSTRACT

FOSTER, GARRETT DANIEL. Expansion of Alternative Generation Techniques. (Under the direction of Dr. Scott Ferguson).

Designing a system for a specific scenario can be a challenge in of itself. However, changes in model fidelity, customer preferences, and desired system architecture further complicate the design process. To respond to this challenge we must move away from the current paradigm that only looks for the single "best" design. Instead, we must search for multiple solutions that can help a designer overcome specification changes that arise throughout a system's design cycle. This thesis investigates the expansion of an existing optimization framework that assists a designer locate and explore design alternatives. Such alternatives exist within a specified performance threshold of an originally selected design. Specifically, this thesis investigates two important questions regarding the process of alternative generation. The first question assesses whether the existing technique can correctly explore the design space in multiobjective nonlinear engineering design problems. From the results, recommendations for improvements are given. Meanwhile, the second research question looks at making the alternative generation computationally viable. The results show that the previously explored designs of a genetic algorithm can be used to increase computational efficiency.

Expansion of Alternative Generation Techniques

by
Garrett Daniel Foster

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Mechanical Engineering

Raleigh, North Carolina

2011

APPROVED BY:

Dr. Gregory Buckner

Dr. Larry Silverberg

Dr. Scott Ferguson
Chair of Advisory Committee

ACKNOWLEDGMENTS

First and foremost I offer my sincerest gratitude to my advisor, Dr. Scott Ferguson, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my Masters degree to his encouragement and effort and without him this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

Further, I thank my committee. I found the questions and comments of both Dr. Gregory Buckner and Dr. Larry Silverberg to be insightful and encouraging.

I also thank my fellow lab members in the System Design Optimization Lab: Micah Holland, Ben Richardson, Eric Sullivan, Marc Tortorice, and Callaway Turner for the interesting discussions and help editing this thesis.

Last but not the least; I would like to thank my loving and supportive wife, Kate Foster. I am eternally grateful for her patience and understanding of all the time I spent in the lab as well as her encouragement throughout the thesis writing process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION AND MOTIVATION

1.1 Benefits of Joint-Cognitive Systems

Imagine going to your favorite search engine. You type in a few keywords and hit the search button. The search engine then returns a link to a single website and nothing else. How confident are you that the search engine has found the best possible website? Thankfully, this is not how modern search engines work. Instead they present the user with an extensive rank-ordered list of websites, normally 10 at a time, and rely on the user's expertise and intuition to choose what is best. This is an example of a joint-cognitive system [1].

A joint-cognitive system combines the strengths of the machine with the strengths of the human user to achieve much greater performance. Machines are very good at doing repetitive tasks, such as sorting designs based on some pre-defined performance metric. However, machines are not very good at understanding the nuisances of how a design performs, as they rely on simplified models that are full of uncertainty. This uncertainty includes the correctness of the assumptions being made, whether important problem factors were initially ignored, and the accuracy of the estimated operating conditions. Conversely, humans are ill suited for doing repetitive tasks, but are very good at understanding the nuisances of how a design performs and are less affected by uncertainties in the system model. Therefore, it is beneficial to use machines to sort and eliminate bad designs, while having the user choose from a few good designs.

Similar to a search engine, engineering design tools are used to search out good designs. However, far too often these design tools over-reach and are asked to choose the single best design. As indicated in the previous paragraph, this is a not the strength of the design tools. Instead, the design tools should assist the designer by showing them a set of designs that perform well to the modeled objectives. One such design tool that does this is called Modeling to Generate Alternatives (MGA).

1.2 History of MGA

The initial development of MGA was motivated by a need to overcome a designer's premature attachment to a single design. This attachment can be attributed to the designer being presented with a single design that algorithm identified as the best. By showing the designer only a single design, the range of viable designs that would normally be considered is drastically reduced. As previously indicated, the strength of a machine is not choosing the best design, but instead filtering out all the bad, or poorly performing, designs from consideration.

MGA development starting in the early 1980's and was initially applied to land use planning problems [2-5]. A side effect of the time at which this approach was developed was that the problems needed to be modeled as linear programming problems to save on the limited computational resources. For example, it cost approximately $9 to run an optimization problem for 30 seconds on the high end computers of the day [4]. Nowadays, it is not uncommon to have optimization problems that take multiple days to converge to a solution. To keep costs down, it was important to identify alternative generation approaches that could efficiently generate small sets of feasible alternatives that performed similarly. Efficiency was measured as a tradeoff between the cost to use the approach and the amount of uniqueness gained by the alternatives the approach identified.

Many possible approaches were explored, and two were shown to be effective at generating alternatives. These two approaches were: (1) algorithms based on random design variable selection and (2) algorithms based on maximizing the difference between design variables. Both approaches operating on largely the same basic principle, which is maximizing the difference between the variables in two different designs while staying within the original constraint of the problem as well as some user defined performance threshold. The two approaches differ in how they calculate the difference. The first approach listed would apply a special type of weighted sum to the variables. Essentially, variables that were randomly selected would receive a weight of '1', while the variables that were not randomly selected would receive a weight of '0' causing them to be ignored during the optimization process. The second approach listed, which was referred to as Hop Skip Jump (HSJ), did not apply any weight to the variables and considered them all equally during the optimization process.

Testing with both approaches showed that they possessed similar performance [4]. However, Kripakaran and Gupta opted to use an approach similar to HSJ to optimize a Moment-Resisting Steel Frame [6]. Their goal was to identify the locations that could minimize the cost of the structure and maintain enough strength to resist wind loading. While, the approach itself was similar to the HSJ approach, the formulation was easier to work with and therefore easier to modify. For this reason it was chosen as the basis for the MGA framework in this thesis.

More recent applications of MGA include the use of genetic algorithms to help solve watershed development problems [7] and an application of MGA to energy modeling [8]. The genetic algorithms were used to start the MGA search process before handing control off to a more efficient algorithm. Meanwhile, the energy modeling used MGA to explore a conceptual model of the U.S. electric sector.

1.3 How MGA Works

Modeling to Generate Alternatives (MGA) is a framework that identifies unique alternative designs that occur within a user defined performance threshold of an original design. This framework (Equation 1.1) allows for optimization techniques to be used, which ensures that the alternative designs are as unique as possible. Within Equation 1.1, the goal is to find design 'b' such that the difference between designs 'a' and 'b' are maximized. However, design 'b' has to fall within the same constraints, 'K_i', as the original problem. It also has to perform within a user defined target performance threshold, 'T_j'. In the rest of this section, the process of performing MGA (Figure 1.1) will be explained in more detail.

$$\begin{aligned} &Max: z(b) = \delta_{ab} \\ &s.t.\ \ g_i(X_b) \leq K_i\ (for\ all\ i) \\ &\quad\ \ f_j(X_b) \leq T_j\ (for\ all\ j) \end{aligned} \tag{1.1}$$

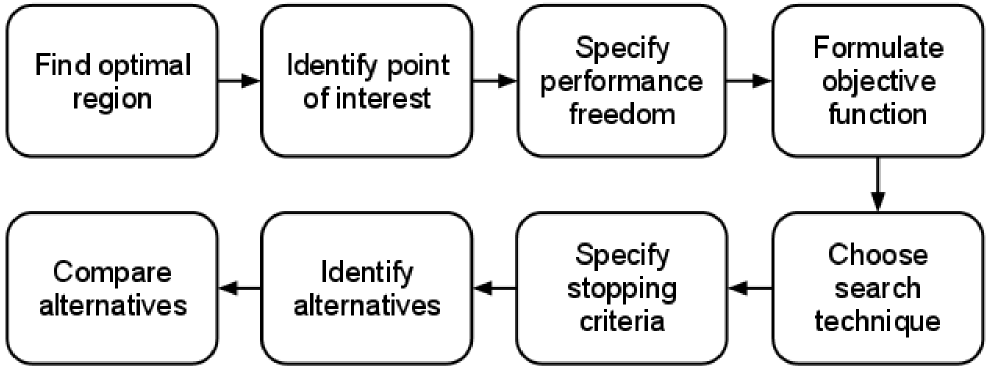


Figure 1.1 MGA Process

1.3.1 Find Optimal Region

The process of performing MGA starts with the identification of the optimal region of the performance space. In previous MGA case studies this optimal region has been a single point. This is a side effect of formulating the problems as single objective, i.e. 1 dimensional, problems. Within this thesis the problems of interest are multiobjective or multi-dimensional. This results in the optimal region being a collection of optimal points. For example, in a two objective, or two dimensional, problem the optimal region will be a curve in the performance space. Likewise, the optimal region in a three objective problem will be a surface in the performance space. These collections of optimal designs are commonly referred to as a 'Pareto frontier'. Methods for finding a Pareto frontier include multiobjective genetic algorithms [9] and weighted sum optimization [10].

1.3.2 Identify Point of Interest

The second step of the MGA process involves choosing a starting design which this thesis refers to as a point of interest. Throughout this thesis, the point of interest lies on the Pareto frontier, however this is not a requirement. The choice of a suitable point of interest is based on the designer's preferences towards each objective. There exist several methods that can assist the designer in choosing a suitable point of interest from the Pareto frontier. Two such methods that help the designer assign an importance or weight to each objective are the house of quality (HoQ) [11] and the hypothetical equivalents and inequivalents method (HEIM) [12-13]. The

HoQ requires that the design define the importance of various customer needs and their relation to each objective. Meanwhile, HEIM has the user choose from parings of hypothetical designs. Each is then able to calculate an objective weighting scheme for the designer. Applying a weighted sum with these weights on the objectives will normally cause a single design on the frontier to evaluate as the best design for the user.

1.3.3 Specify Performance Freedom

The third step requires that the designer define some allowable change from the performance of the point of interest. Within this thesis this allowable change is referred to as “performance freedom”. The performance freedom will be different for each designer as it is based off of their preferences. For example, if the designer were designing a new SUV that achieved ‘good’ fuel economy they might specify ‘good’ as 25-35 mpg. Meanwhile, a different designer might define ‘good’ as 29-31 mpg. As you might suspect, the first designer would specify a larger performance freedom than the second designer. Performance freedom typically starts around 10%-15% of the objectives performance value [14], i.e. a SUV getting 30 mpg would have 3-4.5 mpg of performance freedom.

1.3.4 Formulate Objective Function

The fourth step in the MGA process is to define the objective function that will be maximized. For this thesis the objective function takes the form of a distance calculation between two points (or designs) in the design space. As a result this thesis refers to the objective function as a “distance metric”. Two common distance metrics are used within this thesis: a 1-norm distance and a 2-norm distance. The one norm distance is analogous to the distance traveled walking around a city block, while the 2-norm distance is analogous to distance traveled walking in a straight line. An optional step is to scale the variables of each design prior to calculating the distance between two of them. This scaling can help to reduce bias towards designs that possess a change in a variable that has a larger number of significant digits, i.e. the distance metrics would favor designs that have 1000 mm in change rather than designs that have 1 m in change even though they have an identical amount actual change.

1.3.5 Choose Search Technique

The fifth step involves choosing a technique that will locate the alternative design with the maximum distance measurement. This thesis refers to these techniques as ‘search techniques’. There are two main categories of search techniques investigated in this thesis. The first type uses one design at a time, while the second type uses a population of designs. The one at a time search techniques are characterized by good accuracy to the true optimum. However, depending on the optimization algorithm chosen, they can get stuck at a non-optimal solution or take a long time to converge. Meanwhile, search techniques that use a population of designs are less likely to identify an inferior solution as the best solutions. This can be attributed to population of designs covering a wider range of the performance region, and therefore having a greater opportunity to identify the true optimum. However, the accuracy of these techniques is highly dependent on the sample size chosen. This means high accuracy normally comes with high computational cost.

1.3.6 Specify Stopping Criteria

The sixth step is to define when the designer would like for MGA to stop finding alternative designs. Theoretically, the MGA process can continue for as long as there are unidentified alternative designs. However, it has been suggested that people have a hard time choosing from a large list of choices as they quickly become overwhelmed [15]. Therefore, it is recommended that the designer define some type of stopping criteria. Since MGA is an optimization framework, many of the same stopping criteria used in optimization can be used for MGA. The simplest option is to define an integer value corresponding to the maximum number of alternative designs the designer would like MGA to identify. Alternative options tend to involve setting a threshold for solution quality. For example, a designer may specify that once the decrease in the distance from the first identified alternative design to the most recently discovered alternative design is greater than a certain value the process should stop. Similarly to other optimization frameworks, the designer is allowed to specify more than one stopping criteria at a time.

1.3.7 Identify Alternatives

The seventh step covers the identification of the alternative designs. The search technique starts at the point of interest and seeks out a design that maximized the distance metric while also performing within the designers defined performance threshold and not violating any constraints. Once the stopping criteria is met the search stops and the designer is presented with the most unique alternative designs. It is then up to the designer to use their experience to choose the alternative that is best for their situation.

1.3.8 Compare Alternatives

The final step is used by this research to extract additional information from the alternative searches. Throughout the case studies (Chapter 3 and Chapter 4) this thesis compares different MGA settings. For example, in Chapter 3 both 1-norm and 2-norm distance are compared. Meanwhile, several search techniques are compared to one another in Chapter 4.

1.4 Current Limitations of MGA

Throughout the literature, MGA has been purposely applied to single objective optimization problems. Reasons for avoiding multiobjective optimization problems include computational expense and difficulty associated with setting optimization parameters [6]. Unfortunately, modeling a problem with more than one objective as a single objective problem leads can lead to undesirable results. This is largely due to the optimization relying on the creation of an aggregate objective function using either some convoluted combination of the objective functions or a weighted sum of all the objective values. The latter requires the designer to know the appropriate weights *a priori*. Instead, it would be much more beneficial to expand MGA such that it would be able to handle multiobjective nonlinear engineering design problems as these are the most common in mechanical and aerospace engineering. Also, with multiobjective optimization the designer does not have to specify weights prior to searching or create some convoluted function that has no real world meaning.

When expanding MGA from single objective problems to multiobjective problems several new challenges are introduced. The first of which involves assessing whether MGA can correctly find alternatives in a multiobjective problem. This concern is raised due to the differences found between solving a single objective problem and a multiobjective problem. Unlike a single objective problem, multiobjective problems do not possess a single optimal point. Instead, they will have a region of the performance space filled with non-dominated points called the Pareto frontier [16]. To find a Pareto frontier it becomes necessary to leverage multiobjective optimization approaches such as multiobjective genetic algorithms (MOGAs) [9, 17]. Typically, the objectives in a multiobjective problem conflict and require a designer to make tradeoffs when selecting a final design. When graphed, such as in Figure 1.2, one can identify the non-dominated set of performance points represented by the Pareto frontier. To be part of the Pareto frontier a design must meet the criteria in Equation 1.2. The first criterion states that there cannot be any design that has performance that is better than or equal to the performance of the design under consideration in all objectives. Meanwhile, the second criterion states that there cannot be any design that has better performance in any one objective. If both criteria are met, the point is considered strongly Pareto optimal, however if only the second criteria is met then the point is considered to be weakly Pareto optimal.

$$
\begin{aligned}
f_i(\boldsymbol{x}) &\nleq f_i(\boldsymbol{x}^*) \; for\ all\ i \\
f_i(\boldsymbol{x}) &\nless f_i(\boldsymbol{x}^*) \; for\ any\ i
\end{aligned}
\qquad \textbf{(1.2)}
$$

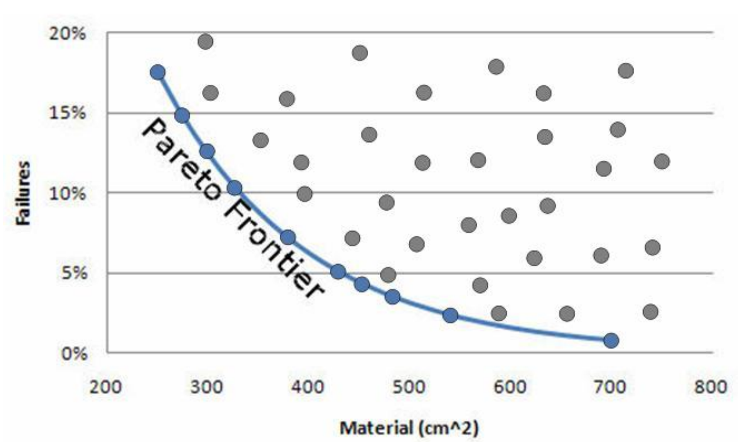


Figure 1.2 Pareto Frontier

Selecting a final design from this frontier now depends on the perceived importance of, or weight associated with, each objective. Some approaches to select this design rely on capturing consumer preferences to determine the region of the Pareto frontier that best meets the demands of the customer [11-13]. Other approaches rely on software tools [ref] for multidimensional visualization as a means of facilitating design steering, often referred to as "Design by Shopping" [18]. In this environment, designers (or consumers) are allowed to simulate a large number of alternatives and actively select regions of the performance space that are of interest to them. Such capabilities provide an initial framework for mass customization [19] in that by selecting a region of the performance space, consumers may now choose a design configuration. This leads to personalized and customized products that may also be unique; a factor that has been shown to drive the market for customized products [20].

A second challenge is reducing computational cost. As mentioned, one reason for not using MGA on multiobjective problems is the high associated computational cost. This increase in computational cost occurs in both the solving of the Pareto frontier and the application of MGA. Additionally, if these problems are non-linear they will likely take even longer to evaluate. It is important to keep the cost of performing MGA low; otherwise the designer would be deterred from using MGA, which could lead to an inferior design being chosen.

Beyond these additional challenges remain, such as the appropriate settings for performance freedom and stopping criteria. However, these challenges are not unique to the application of MGA to multiobjective problems. Instead this thesis will focus on overcoming the first two challenges, which are likely to be affected by the application of MGA to multiobjective problems.

1.5 Research Questions

To facilitate the expansion of MGA into a design tool that can handle multiobjective nonlinear engineering design problems, two main research questions will be explored. The first question is concerned with making the appropriate changes to the current MGA technique such that it will produce useable results in a multiobjective nonlinear problem. The second question is concerned with identifying search techniques that allow MGA to be performed in a computationally efficient manner.

1.5.1 Research Question #1

How does the definition of the objective function affect the alternative designs found by MGA in a multiobjective nonlinear engineering design problem?

In Section 1.3 several settings that go into the problem setup portion of MGA are identified. These include the choosing of the point of interest, specifying the performance freedom, defining a distance metric, choosing a search technique, and specifying stopping criteria. It is expected that most of these will effect which alternatives are discovered. However, some are expected to affect the results more than others, especially in a multiobjective nonlinear engineering design problem.

For instance, the main objective of the MGA process is to maximize a distance. Therefore, it is expected that the choice of distance metric will play a large role in determining which alternatives are found. Many of the existing case studies used either 2-norm distance or a weighted sum to arrive at a distance measure. However, 2-norm distance may not be appropriate when the design variables have different units. Conversely, if appropriate scaling is applied then the variables could essentially be considered unit-less. Additionally, a 1-norm distance metric could be more robust than a 2-norm distance metric. The reasoning is that 2-norm distance squares the difference between two variables. This causes an increase in the bias towards variables with larger magnitudes. However, 1-norm distance uses only addition and therefore does not increase the bias towards variables with larger magnitudes.

Additionally, it is expected that different search techniques will identify different alternative designs. This is similar to the way that different optimization algorithms return different optimum designs for the same problem. Reasons for finding different alternative designs included using a search that gets stuck near locally unique alternative designs or the use of search technique with poor accuracy or fidelity.

To investigate how much, if any, distance metrics affect the alternative designs identified; MGA should be applied to a multiobjective nonlinear case study problem. To eliminate the assumed effects of the search techniques, a single robust search technique should be chosen. Additionally, the point of interest, the performance freedom, and the stopping criteria should be kept constant as these should also have an effect on the alternatives identified. Controlling these settings allows the effects of the two distance metrics to be investigated independent of the effects of the other settings.

1.5.2 Research Questions #2

Can graveyard data be used to perform MGA in a computationally efficient manner on multiobjective nonlinear engineering design problems?

Recall that one of the major reasons for keeping MGA relegated to single objective problems has been the high computational cost associated with multiobjective problems. However, if the computational cost can be reduced then to an acceptable level then there is one less barrier for using MGA on a multiobjective problem. The level of acceptable computational cost is somewhat arbitrary, but largely depends on the number of function calls needed to arrive at a solution. The more functional calls, the more time a process takes regardless of processor speed. As a bare minimum the number of calls needed to find a single alternative should be less than or equal to the number of functional calls it took to find the point of interest. If it takes longer to find the first alternative than it does to find the point of interest, then the designer would likely forgo MGA altogether.

This thesis hypothesizes that if the designs evaluated during the search for the Pareto frontier are retained in a 'graveyard', that the computational cost to find alternatives can be dramatically reduced. To test this hypothesis, search techniques that make use of the graveyard data should be compared to search techniques that do not make use of graveyard data. Both types of search techniques (this with graveyard and those without) should be tried on multiple multiobjective nonlinear engineering design problems to ensure results are broadly applicable. Additionally, to eliminate the effects of any other settings, the point of interest, performance freedom, distance metric, and stopping criteria should be kept the same. Further, the accuracy of the various search techniques should also be measured. It is expected that the accuracy of a search technique will have some correlation with the computational cost of the search technique, i.e. low cost low accuracy. This tradeoff is of great importance to the designer. For example, some simulations take several hours to several days to run. In these cases accuracy could be compromised for computational costs. Conversely, in simulations that run in as little as a few seconds the computational costs are likely of no importance and the designer would instead opt for higher accuracy.

1.6 Thesis Organization

To investigate the two questions outlined in section 1.5, this thesis is divided into five chapters. Chapter 1 gives an introduction into the motivation behind this research as well as introduces the research questions that this thesis addresses. Chapter 2 presents background into research that is related to MGA as well as indentifies a few additional benefits of MGA. Chapters 3 and 4 explore the research questions. Chapter 3 specifically looks at the first research question, while chapter 4 is dedicated to exploring the second research question. Both chapter 3 and chapter 4 are broken up into four sub-sections. The first sub-section restates the design question and the motivation for it. The second sub-section describes the method used to investigate the question. The third sub-section presents the application of the method on two different case study problems. The fourth and final sub-section discusses the results from the case studies. Finally, chapter 5 concludes this thesis with a summary of the results and areas of future work.

CHAPTER 2: BACKGROUND

2.1 Design by Shopping

As stated in the previous chapter the goal of this research is to extend MGA so that it is better suited in handling multiobjective, nonlinear engineering design problems. Again, these types of problems are more prevalent in both mechanical and aerospace engineering. Within multiobjective problems the designer is presented with a range of designs that they can choose from. This range of designs occurs as a result of no single design being able to dominate all other designs in terms of performance. Instead, multiple designs are needed to represent the tradeoffs from one objective to another. However, the current approach suggested by MGA to deal with multiobjective problems is to combine the objectives into a single performance metric, either through a weighted sum approach or by defining some value function. This shift in paradigm from looking for a single design to instead looking for a range of designs is referred to a "design by shopping". According to Balling [18], design by shopping allows the designer to visualize the design space before selecting a final design. This allows the designer to delay the formation of their design preferences until later in the design process. Conversely, the current MGA process requires that the designer specify their preferences *a priori*. This idea of delaying design decisions also shows up in design tools such as functional modeling [21] and set-based design [22-25].

2.2 Related Research

MGA is not alone in the quest to present the designer with multiple good designs. These related approaches fall loosely into three categories. Those categories are overcoming modeling uncertainties through the use of a join-cognitive system, decreasing the convergence time in concurrent design by identifying a range of solutions, and locating design freedom. These three categories are discussed in further detail below.

2.2.1 Model Uncertainty

As previously mentioned, engineering design models are full of uncertainty. This uncertainty comes from many sources. Examples of this uncertainty include assessing if the correct assumptions are being made, exploring if important problem factors were initially ignored, and questioning the accuracy of the estimated operating conditions. If there is a problem with any of these, the design decision tools will likely identify a non-optimal, or infeasible, design as being the best. In the rest of this section an introduction to a few approaches that attempt to overcome model uncertainty are presented.

Wood [26-28] used interval mathematics and α-cuts from fuzzy calculus to overcome parameter imprecision. His work was applied to realistic engineering functions and allowed the designer to perform engineering design calculations that were subject to some level of uncertainty. The results indicated that imprecise calculations could be used with these functions at reasonable computational costs.

A second family of approaches that attempts to overcome partially defined or ill-defined problems is evolutionary algorithms. They operate on a similar concept as MGA, in that they try to present a range of solutions to the designer. They accomplish this goal by maintaining a diverse population of solutions. Algorithms that fall under this heading include evolution programming [29], evolutionary strategies [30], genetic algorithms [31-32], and genetic programming [33]. Xiao et al. applied evolutionary algorithm approaches to site-search problems [34]. Their work builds upon the development of robust evolutionary algorithms that are focused on alternative generation [35]. The results from these indicate that evolutionary algorithms can be used to generate alternative solutions that can assist with multiobjective decision making.

Additional approaches to dealing with uncertainty include the forecasting or prediction of parameter uncertainty [36] and interval modeling [37]. Möller [38] provides a good overview of how uncertainty in engineering is currently approached.

2.2.2 Concurrent Design

The next group of approaches relate to MGA's ability to generate a range of solutions that are suitable for improving the convergence time of concurrent design. It is hypothesized that passing back a range of designs could to improve convergence time in collaborative design environments, such as the design of an automobile. Currently, it takes several years for an automobile to go from the drawing board to production. To speed up this process, more of the development has to take place concurrently. However, each designer has their own set of goals and priorities. Subsequently, if two designers pass back only a single best solution it is unlikely that the solutions will overlap. This leads to an increased number of iterations to arrive at a final solution as the designers barter with one another. Alternatively, if the designers passed back a set or range of solutions the likelihood of overlap increases, thus decreasing the needed iteration.

Set-based design approaches [22-25] have been used in concurrent engineering environments as a means of maintaining design freedom. Motivation for this approach is to delay the need for making design commitments until later in the design process, allowing for increased response to uncertainty and changing customer needs. Recent work by Madhavan et al. [39] has shown that in an industrial setting, set-based design approaches reduce the number of iterations between design teams and provide a library of back-up design options.

Physical Programming methods developed in [40-41] differentiate solutions into five different types based on the quality of the solution, making all solutions within the same type equivalent. In a different vein, identification of alternative designs through the application of decision trees has been studied by Malak and Paredis [42]. Multiobjective Genetic Algorithms (MOGAs) that seek Pareto-optimal solutions while including solution diversity as an additional objective can also be used to increase freedom present in the final design [43-44]. However, much of the research within MOGAs is focused on reducing the computational overhead needed to find the Pareto frontier [45-46]. Instead, this paper looks to increase the usefulness of the data generated by a MOGA.

2.2.3 Locating Design Freedom

The final group of approaches is focused on identifying design freedom. As previously mentioned, points of interest with high design freedom are candidates for mass customizable systems. These areas would allow a designer to offer a custom experience to their customers without a substantial change in performance.

One of the simplest approaches to finding design freedom is classical sensitivity analysis [47-48]. Sensitivity analysis can tell the design where to look for design freedom, but it cannot tell the designer what design variables make up the alternatives. Beyond this, target- or goal-seeking algorithms have commonly been used as an approach for finding unique designs with similar performance [49]. For these algorithms, the objective functions are minimizing the distance from a target performance while maximizing the distance in the design space between solutions. MGA is one such target-seeking framework.

Simov conducted an in-depth study into the significance of design freedom or as he refers to it, one-to-many mappings [50-51]. The study is applied to four engineering case study problems. The results from the studies indicated that design freedom was not very common near the Pareto frontier. Further, the approach used to identify regions of design freedom was shown to be sensitive to the allowed performance region.

2.3 Benefits of MGA

As the previous section indicates there are several approaches that aim to accomplish similar goals as MGA, however MGA offers many benefits that the other approaches lack. First, recall that design by shopping makes use of visualization to help the designer form better design preferences. Consider that it is difficult plot and visualize anything greater than three dimensions. Thus, the current research into visualization techniques looks at ways to show the data when it exceeds these limits. Of the related approaches in the previous section, MGA is one of the few that would provide new data for these visualization techniques. To better understand this concept, consider that most of the techniques are striving to find optimal designs. Conversely, MGA is allowing the designer to find non-optimal designs in addition to the already found optimal designs. Ideally, the designer could choose any point from the visualization technique and MGA would then locate new designs for the designer to consider. As already mentioned these designs would perform similarly to the designer's point of interest.

A second benefit of MGA exists is that it can be applied to already solved problems. In essence, the designer can begin by using whatever optimization or design tools they are comfortable with to find a starting design. Once they find something they are interested in they can enlist the help of MGA to identify similarly performing unique designs. This reduces the learning curve required for its use. Further, this means MGA can be applied to existing data sets and doesn't require that they be completely resolved. If the designer doesn't like the results from the MGA approach, they can go back to their previous methods as MGA doesn't modify the model or subsequently, the data it uses.

Additional benefits are expected in the identification of design freedom. Currently, the approaches used to identify regions with good design freedom require large computational costs as they have to sample a large portion of the design space. MGA should be able reduce this computational burden by allowing the designer to pick and choose where they want to investigate for design freedom. Additionally, the process could be automated to show optimal regions of the performance space that possess relatively good design freedom.

A final benefit of MGA versus similar approaches is that it should be relatively easy to customize. This idea of being able to "tune" an algorithm to the type of problem that it is working on is common in optimization algorithms [9]. For example, a designer might include a pre-defined bias towards a certain variable that they know is easier to change. In this way MGA is more likely to identify unique designs that have changes in this variable dimension. Additionally, it is expected that users could trade accuracy for computational costs. Consider the case where it takes several hours to evaluate a single design. For this scenario the designer might choose to relax the needed accuracy of MGA to gain a reduction in the computational expense.

2.4 Supporting Research

To support the expansion of MGA this thesis makes use of other research. This research includes multiobjective genetic algorithms, the Nelder-Mead simplex method, latin hypercube sampling, grid search, and statistical analysis of variance. A brief discussion of each is included below.

2.4.1 Multiobjective Genetic Algorithms

Multiobjective genetic algorithms (MOGAs) are an optimization approach used to identify a set of non-dominated set of performance points, commonly referred to as the Pareto frontier [16]. While a MOGA may be individually tailored to a specific problem, all instances have a similar procedure. That procedure can be seen in Figure 2.1. Additionally, MOGAs are part of the evolutionary algorithm family. As mentioned in Section 2.2.1, evolutionary algorithms are good at maintaining a diverse population of designs. This makes MOGAs a good starting point for using MGA with multiobjective problems. Additionally, MOGAs evaluate a large number of designs, which makes it a good candidate for producing both the Pareto frontier and the graveyard data. A downside to evaluating a large number of designs is the increased computational expense. It is up the designer to determine the best tradeoff between computational cost and population size for their situation.

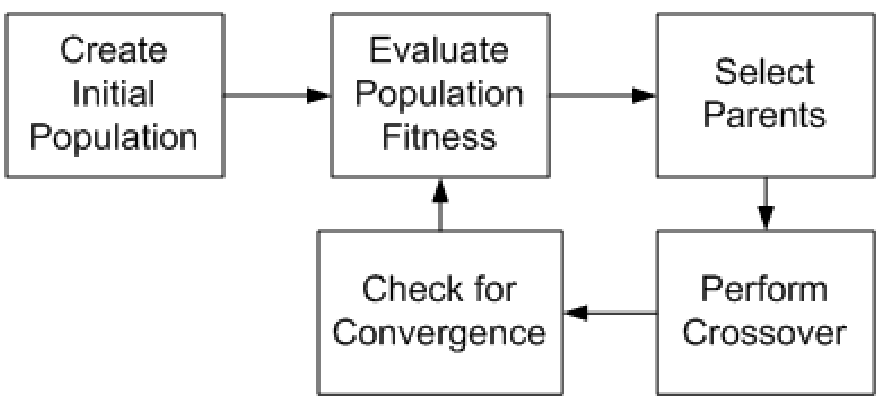


Figure 2.1 Generic Multiobjective Genetic Algorithm Process

2.4.2 Nelder-Mead simplex method

The Nelder-Mead simplex method (NM) is a derivative free optimization method. Since NM does not need to calculate derivatives, it can be performed at much lower computational cost than optimization methods that require derivative information. This reduction in computational cost is one of the main reasons it was chosen for study in this thesis. NM starts by constructing a simplex with n+1 points, where n is the number of parameters. Thus, for a two parameter problem there will be three points that form a triangle. Then a weighted sum of squares is computed at each point. The point with the highest value, assuming minimization, is reflected through the centroid of the simplex. Following this a weighted sum of squares is computed for the new point. If the value of the new point is the lowest of them the process continues by reflecting the point with the highest value. However, if the value of the new point is higher than the other points the simplex is compressed and the new point is reflected closer. If the value of the new point is neither the lowest nor the highest then start at the top and reflect again. These are repeated until the simplex converges. A visualization of the Nelder-Mead simplex method can be seen in Figure 2.2.

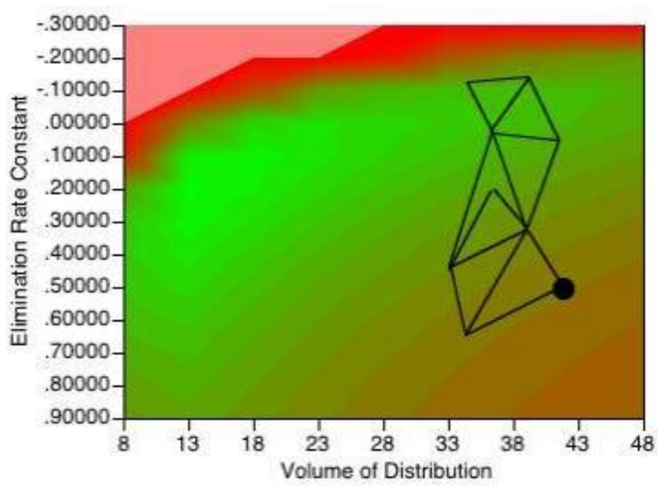


Figure 2.2 Visualization of Nelder-Mead Simplex Method [52]

2.4.3 Latin Hypercube Sampling

Latin hypercube sampling is a stratified sampling technique used to generate a population of distributed designs in problems with multiple variables. To perform the sampling over *n* variables, the range of each variable is divided into *m* equally probable intervals. Each variable has the same number of intervals. For each variable a probability is randomly chosen from each segment and mapped to the actual value according to the variable's distribution. Then the random value in each segment is combined in a random manner with the random values of the other variables to produce designs. For example, a problem with 2 variables and 4 designs would produce 24 possible design combinations. The algorithm used to generate the latin hypercube samplings for this research is lhsdesign [53], which is part of the MathWorks Matlab software package.

2.4.4 Grid Search

A grid search is a fairly simple optimization algorithm [54]. It divides the design space into equally spaced sections. In the case of a two variable problem the design space looks like a grid, hence the name. At the intersection of each section the corresponding value is calculated. Overall, the grid search method is very robust. However, it is also very inefficient as a large number of inferior points are evaluated, hence the reason it is sometimes referred to as an exhaustive search. This thesis makes use of the grid search to ensure that at least one of the search techniques will find the best possible alternatives, albeit at very high computational cost.

2.4.5 Statistical Analysis of Variance

Statistical Analysis of Variance is used to test if all samples in a population are statistically equivalent. More specifically, the type of analysis used is a one way analysis of variance (ANOVA) [55]. An ANOVA is chosen over a standard t-test as it allows for all the candidates to be tested at once, whereas a t-test only allows for two candidates to be compared at a time. The process is a multi-step process that involves the calculation of the sample mean, the variation about the sample mean, the error of individual samples and their corresponding means, computing the average sample variation, computing the average sample error, and computing a test statistic. This test statistic is used to find a corresponding probability that the hypothesis is true. The hypothesis tested in this thesis is that all the search techniques are statistically equivalent. If the hypothesis is shown to be false, the confidence intervals for each search technique can be calculated and plotted. For this thesis a 95% confidence interval is used.

2.5 Chapter Summary

This chapter presents the reader with a background in MGA and related research. To help the reader understand the state of the art in MGA, a brief history is presented explaining its transition from original motivation to the current work that this thesis builds on. Next, similar research was presented demonstrating other approaches that are striving to accomplish the same goals as MGA. Again, these goals overcoming modeling uncertainties through the use of a join-cognitive system, decreasing the convergence time in concurrent design by identifying a range of solutions, and locating design freedom. Following this, background needed to understand the case studies in Chapter 3 and Chapter 4 was presented. This included brief descriptions of optimization algorithms, random sample generation techniques, and a primer on one way analysis of variance.

In the next chapter, the first research question will be explored. Two engineering design problems are used to investigate the effects of distance metric choice as discussed in section 1.5.1. These engineering design problems include the design of a two variable tripod and a six variable vibrating platform.

CHAPTER 3: EFFECTS OF OBJECTIVE FUNCTION FORMULATION

3.1 Motivation

Recall that there are numerous settings that go into the problem setup portion of MGA as explained in Section 1.3. Again, these settings include point of interest selection, performance freedom specification, objective function formulation (i.e. distance metric choice), search technique choice, and stopping criteria selection. It is expected that changing from a single objective linear engineering design problem to a nonlinear multiobjective engineering design problem will alter the way these settings affect the problem. The goal of this chapter is to answer research question #1, which asks "How does the definition of the objective function affect the alternative designs found by MGA in a multiobjective nonlinear engineering design problem?"

As per Section 1.3.4 the objective function is defined as a distance measurement for the purposes of this thesis. Distance is chosen as it represents a scalar value that most users have an easy time visualizing and understanding. Other more abstract methods include variation measurements and random variable weighting schemes. In prior research [6] a hamming distance was used. Unfortunately, this distance metric only measures the difference between two binary strings. Since the variables within the nonlinear multiobjective engineering design problems are unlikely to be represented by binary values a different distance metric is required. Additionally, it is hypothesized that there is a high degree of interaction between the distance metric and how the variables are scaled. As a result, this chapter focuses primarily on identifying the effects of different distance metrics and scaling techniques.

3.2 Method

To investigate the effects of different distance metrics and scaling techniques, the remaining settings must be held constant. This means all searches within a single test case will start with the same point of interest, keep the design freedom constant, utilize the same stopping criteria, and use the same search technique. Recall that MGA has been primarily applied to single objective nonlinear problems as these require lower computational cost, however this paper would like to expand MGA to problems that are both multiobjective and nonlinear. To separate the effects of the problems becoming nonlinear and multiobjective two different problems will be tested. The first will be a single objective nonlinear problem. This eliminates any coupling that occurs between of making the problem both nonlinear and multiobjective. Additionally, this single objective nonlinear problem serves as a simple learning example that helps to further illustrate the MGA process. Meanwhile, the second test problem will be nonlinear and multiobjective as that is the type of problem that this research is interested in. For each test problem, the same series of steps as outlined in Section 1.3 will be followed. For convenience of the reader, Figure 1.1 is re-posted below. The settings for each block in Figure 3.1 are described in the rest of this section.

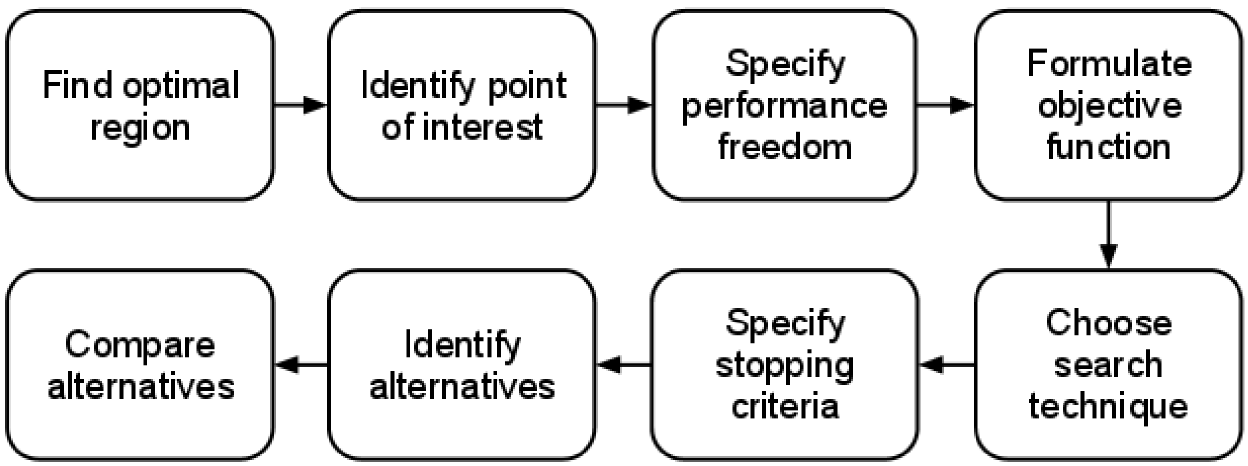


Figure 3.1 Research Question #3 Method Flow Chart

3.2.1 Find Optimal Region

To reduce computational overhead, the optimal region for the multiobjective problem is found using a grid search. This allows the data in the grid search to be re-used later in the MGA process. The size of the grids were set to 1% step sizes or lower to ensure good accuracy of the results. The single objective problem did not require solving for an optimal region as it was supplied in the problem definition [56].

3.2.2 Identify Point of Interest

Once the optimal region was identified points of interest were chosen. For the single objective problem the provided optimum served as a convenient point of interest. Meanwhile, for the multiobjective problem, three points of interest were chosen from the Pareto frontier. These three points represented different regions of the frontier and gave some assurance that the results were repeatable and not unique to a single point or region of the performance space.

3.2.3 Specify Performance Freedom

To allow for unique alternative designs to be identified, the next step relaxes the required performance of the alternative designs. Performance freedom for the two case study problems was chosen as 10% of the point of interest's performance value and represents a fairly common starting range for MGA. This means that the single objective, 6.6 kg, tripod has a performance freedom of 0.66 kg. Further, this means that any feasible alternative has to have a mass that is less than or equal to 7.26 kg since the goal is to minimize mass. If the goal were to maximize mass the 0.66 kg performance freedom would be subtracted from the 6.6 kg optimal value. As you may have noticed, the single objective problem has only one performance freedom value. This is due to there being only one performance axis to consider.

However, for multiobjective problems multiple performance freedom values are needed. More precisely there will be a unique performance freedom value corresponding to each objective. It is not expected that the value of the performance freedom for one objective is always equal to the performance freedom of the other objectives.

3.2.4 Formulate Objective Function

As mentioned previously, maximizing the objective function is what drives the MGA process. As a result the objective function is likely to play a large role in which alternative designs are identified. A convenient way to define the objective function is through the use of distance metrics. As mentioned, distance metrics are easier to visualize and comprehend than other objective function formulations. The two different distance metrics that will be considered include both a 1-norm and a 2-norm distance metric [57]. The 1-norm distance metric, which is sometimes called a Manhattan distance [58], is shown in Equation 3.1. It is considered because it should prevent further bias from being introduced into a poorly scaled problem. This subsequently decreases the importance of variable scaling, which makes MGA easier to setup. To illustrate this idea, consider the numbers 1 and 100. If these numbers represent the differences between variable 1 and variable 2, respectively, then the total distance according to a 1-norm distance is 101. Now, if we change the values to 11 and 90, the distance is still 101. Meanwhile, if we apply the same situation to two norm distance we get distances of approximately 100 and 90.7. From this simple example it is easy to see how the two norm distance can induce additional bias into the problem. The 2-norm distance metric, which is based on Euclidean distance, is shown in Equation 3.2. It is considered because it is the most common distance metric used in mechanical and aerospace engineering design problems. Additionally, it is easier to use in vector based math. In both Equation 3.1 and Equation 3.2, subscript *a* denotes the first design, subscript *b* denotes the second design, c_i denotes the ith design variable of the respective design, and d_{ab} denotes the distance from design a to design b.

$$\boldsymbol{d_{ab} = \sum_{i=1}^{n} |c_{ia} - c_{ib}|} \qquad \textbf{(3.1)}$$

$$\boldsymbol{d_{ab} = \sqrt{\sum_{i=1}^{n} (c_{ia} - c_{ib})^2}} \qquad \textbf{(3.2)}$$

To reduce the tendency of either distance metric to choose alternatives containing variables with larger magnitudes, the design space needs to be scaled so that all variable axes are biased according to the designer’s preferences. The scaling techniques chosen for investigation are no scaling, percent difference scaling, bound scaling, and max step scaling. The first option is the choice to forgo scaling the variables. This can safely be done in cases where the variable ranges are of similar magnitudes. This also, makes the problem easier to setup and understand. The single objective problem does not use scaling for these reasons.

The first actual scaling technique is percent difference scaling. Percent difference scaling is considered as it can be applied to any problem that exists on a continuous range; however it will produce undesirable results if any variables are close to zero in value. This behavior is easier to understand if one considers the formulation for percent difference scaling (Equation 3.3). If either variable in the equation are near zero, the equation will return a value of 2 regardless of the value of the other variable. Additionally, percent difference scaling does not provide for uniform scaling as seen in Figure 3.2. This creates a disadvantage for larger variables.

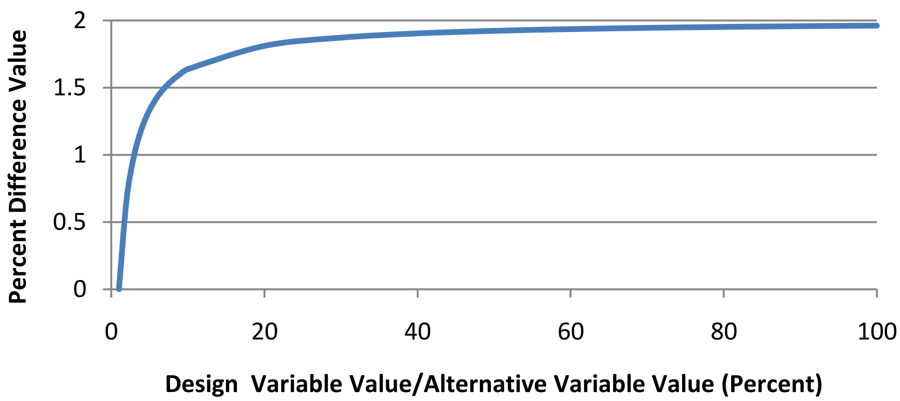


Figure 3.2 Non-Uniform Scaling of Percent Difference Scaling

The bound scaling approach is also referred to as normalization. It provides a more uniform scaling to the variables in a problem than percent difference scaling. If plotted like Figure 3.2, the scaling would form a straight line. However, it requires that the variable bounds be known or specified beforehand. Its formulation can be seen in Equation 3.4.

The last of the three scaling techniques is the max step technique. Max step was created to try to combine the advantages of the percent difference scaling and the bound scaling. Max step can be applied to problems without specifying bounds beforehand. As the name suggests, max step will identify the range of the variables that are feasible within the specified performance freedom region. It will then scale the variables in each alternative according to this range. The downside to this approach is the added computational cost as it has to identify the range of variable values. The formulation for this technique can be seen in Equation 3.5.

$$\text{Percent Difference Scaling} = \left(\frac{\text{design variable value} - \text{current alternative variable value}}{\frac{\text{design variable value} + \text{current alternative variable value}}{2}} \right) \tag{3.3}$$

$$\text{Bound Scaling} = \frac{\text{design variable value} - \text{current alternative variable value}}{\text{variable upper bound} - \text{variable lower bound}} \tag{3.4}$$

$$\text{Max Step Scaling} = \frac{\text{design variable value} - \text{current alternative variable value}}{\text{variable range upper bound} - \text{variable range lower bound}} \tag{3.5}$$

3.2.5 Choose a Search Technique

The search technique chosen for both the single objective and multiobjective problems is a grid search. Again, the grid search in each problem is performed at a fairly fine interval, less than 1% of the provided design variable bounds. The grid search is chosen as it provides results that are repeatable along with good accuracy considering the step size chosen. Since this chapter is focused on distance metrics and scaling techniques it is important to have a repeatable results for making accurate comparisons. To help decrease the computational cost the grid search was only performed once for each case study. The results from the grid search were stored in a database so that they could be post process later according the distance metric and scaling technique chosen.

3.2.6 Specify Stopping Criteria

The stopping criteria will be fairly simple for this research question. In both the single objective and multiobjective problems, five alternatives will be created. Generating five alternatives was chosen arbitrarily, but will serve to demonstrate the complexity and challenges that potentially arise when using this approach. In real-world application, there is the challenge that too many alternatives will make it difficult for the designer to focus on any one design. Conversely, too few alternatives will provide very little insight into the problem.

3.2.7 Identify Alternatives

The next step is to find five alternatives using the previously specified settings. However, a distance metric can only be used to find a single alternative in its base form. To find multiple alternatives the prior research suggests adding the distance measures to each successive alternative. For example, when searching for the second alternative, the designer would add the distance to the point of interest with the distance to the first alternative. Theoretically the design chosen as the second alternative would have the largest combined distance. For the single objective case study there are two sets of alternatives, or one for each distance metric. Meanwhile, for the multiobjective case study, there are six possible combinations of the two distance metrics and three scaling techniques.

3.2.8 Compare Alternatives

The final step for each case study is to compare the alternatives found using the previously specified settings. To help illustrate the MGA process, the alternatives in the single objective problem are plotted in a two dimensional design space. This also allows for easy comparison. For the multiobjective problem, the designs are plotted on radar plots as there are more than three variables. In the next section this method will be applied to two different case studies, a single objective engineering design problem and a multiobjective engineering design problem.

3.3 Case Studies

Within this section two case studies will be used to investigate the effects of different distance metrics and scaling techniques**.** The first problem investigated is single objective, while the second problem is multiobjective. Again, this is done to decouple the effects of nonlinear problems from the effects of multiobjective problems.

3.3.1 Tripod

The selected single objective nonlinear problem is the design of a tripod. This problem is adopted from Arora [56]. The objective of the problem is minimizing the mass of the tripod. It must do this while maintaining enough strength so that the tripod will not fail under a 60kN vertical load. Additionally, size constraints are imposed on the tripod's design. The full formulation of the problem can be seen in Equation 3.6 below.

$$
\begin{gathered}
Min: f(D,H) = 3\pi\rho L\left(\frac{D}{2}\right)^2 \\
S.T.\ \left(\frac{P}{L}\right) - \sigma_a \geq 0 \\
P - \left(\frac{P_{cr}}{2}\right) \geq 0 \\
0.005m \leq D \leq 0.5m \\
0.5m \leq H \leq 5m \\
L = \left(H^2 + \frac{1}{2}B^2\right)^{0.5} \\
o_a = 150MPa \\
E = 57GPa \\
W = 60kN \\
B = 1.2m \\
\rho = 2800\,\frac{kg}{m^3} \\
P_{cr} = \frac{\pi^2 EI}{L^2} \\
I = \pi D^4/64 \\
P = WL/3H
\end{gathered}
\qquad \textbf{(3.6)}
$$

The optimal solution to this problem serves as the point of interest. According to the text [56], the optimal solution has a height of 0.5 m and a diameter of 0.0342 m, leading to a minimized mass of 6.6 kg. If the optimum solution had not been supplied, any number of single objective optimization methods could have been utilized to find it.

Applying a 10% performance freedom to this problem means that all feasible alternative designs must have a mass less than or equal to 7.26 kg. To search the rest of the design space a grid search was used. A .001 m step size was used for both variables. This resulted in over 20,000,000 designs being evaluated. Then, when these designs were filtered to include only the population of designs that were less than or equal to 7.26 kg, it was found that they formed a straight line in the design space (Figure 3.3). From Figure 3.3, it is immediately noticeable that only the height of the tripod is able to be change, whereas the diameter remains fixed for all alternatives.

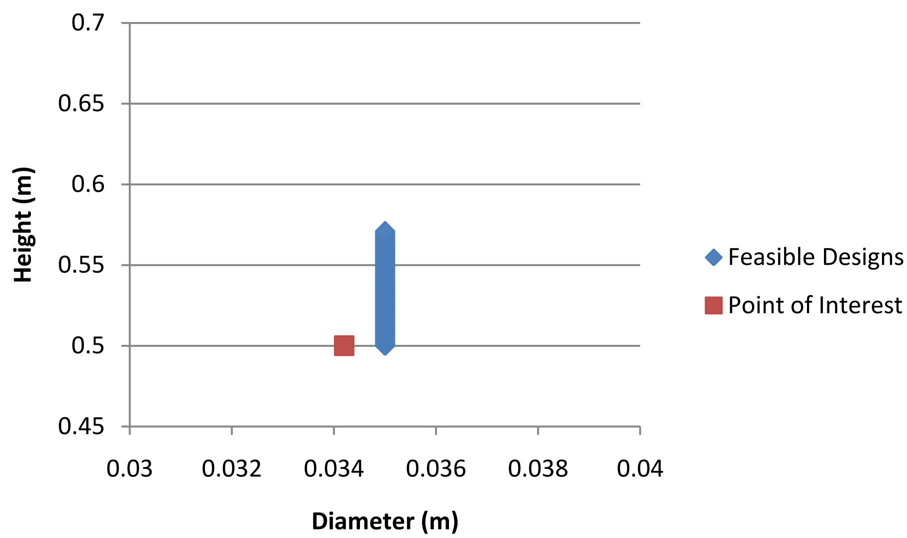


Figure 3.3 Feasible Designs for 10% Performance Freedom

The next step applied both the 1-norm and the 2-norm metrics to the feasible design region to find the five best alternatives. The results (Figure 3.4) show that both alternatives seem to find the same initial alternative and the same second alternative. Then it appears as if no additional unique alternatives are discovered. However, by looking at the label on the points in Figure 3.4, a better picture of how the search progressed can be seen.

After careful examination, it is discovered that the summation of the distance metrics create a situation where the best alternatives occur only at the extremes of the feasible design space. It is possible that this is how the version MGA that this research was based on was intended to work. However, it may be more desirable to have MGA identify a distributed range of designs and not just the extremes of the feasible space. To examine if this result was an artifact of a small performance window and not the formulation of the objective function, the performance variation was increased to 20%. This leads to a broader design space as seen in Figure 3.5.

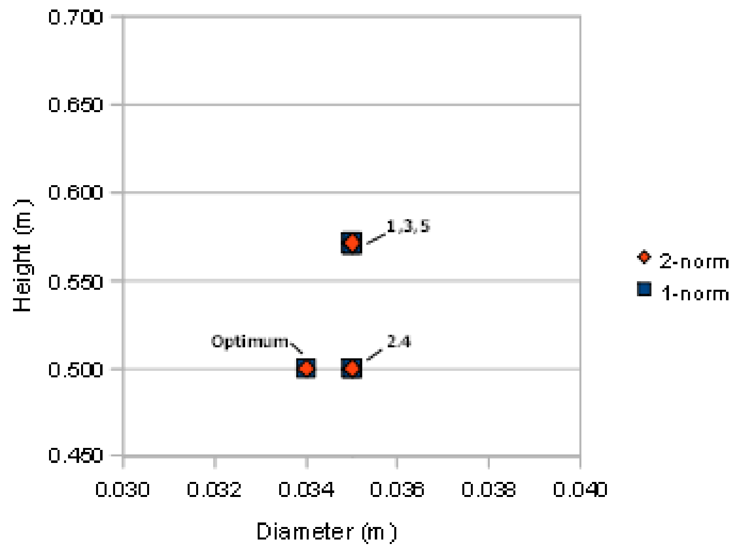


Figure 3.4 Alternatives using Summation Approach with 10% Performance Variability

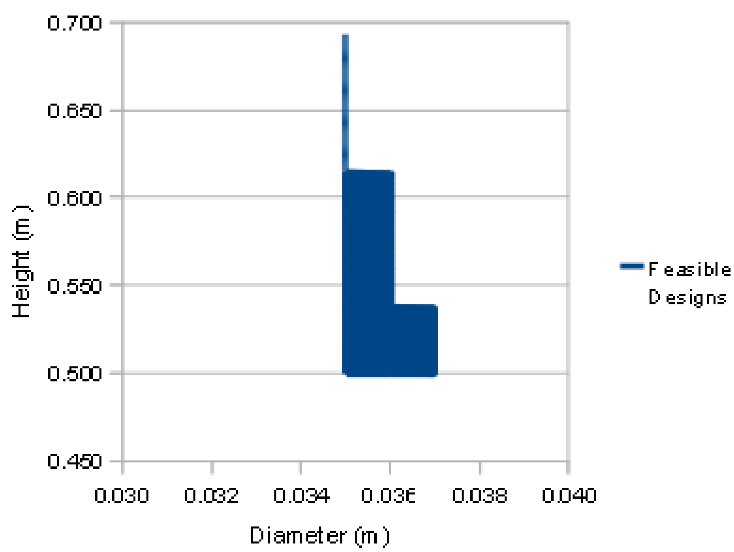


Figure 3.5 Feasible Designs with 20% Performance Variability

Application of the same summation approach to the feasible design space in Figure 3.5 again resulted in no alternatives being found in the interior of the feasible design space. Instead, once the corners of the space were found, the generated concepts alternated between already discovered designs, as seen in Figure 3.6.

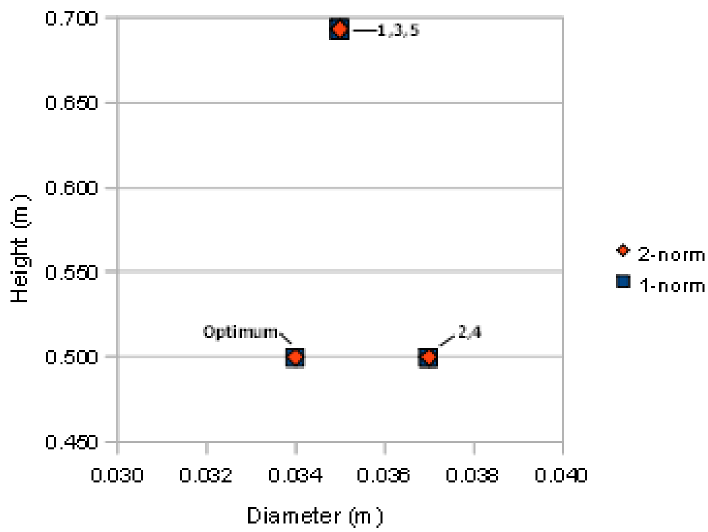


Figure 3.6 Alternatives Using Summation Approach with 20% Performance Variability

This result seems to confirm the notion that the MGA framework that this research builds on is designed to find only extreme points in the feasible design region. In an effort to provide a more uniform representation of the design space, conditional statements that allowed generated alternatives to be discovered only once were created. This essentially removed designs from the pool of potential candidates once they were identified as alternatives. This solution was attempted as an "easy" fix to the challenge, and five unique alternatives were discovered. Unfortunately, the designs still clustered near the corners of the design space.

A second attempt to reduce the clustering of alternatives at the corners of the feasible design space was made. This attempt measured the distance to each alternative separately and used only the shortest distance as the overall distance for a design. To illustrate this idea, consider a number line from 0 to 3. On the number line 0 represents the point of interest and 3 represents and the 1st identified alternative. If the design under consideration for the 2nd alternative is located at 1, then it is 1 unit away from the point of interest, and 2 units away from the 1st alternative. However, it would be represented to the objective function as having a distance of 1, as this is the smallest distance measured (from the design to the point of interest). Effectively, this new approach creates a scenario where designs with zero distance correspond to existing alternatives. This search more naturally tries to push away from existing alternatives, and create a more uniform distribution of alternative designs. The results using this new approach for 10% and 20% performance freedom are shown in Figure 3.7 and Figure 3.8, respectively.

The results in Figure 3.7 correspond to a scenario where the generated alternatives are in a straight line because the allowable performance variation is defined at 10%. The first two alternatives generated occur at the endpoints, and each subsequent design is between two existing alternatives. Figure 3.8 shows the distribution of alternatives using 20% performance freedom. It is interesting to note that both the 1-norm and the 2-norm alternatives were identical considering the increased design freedom.

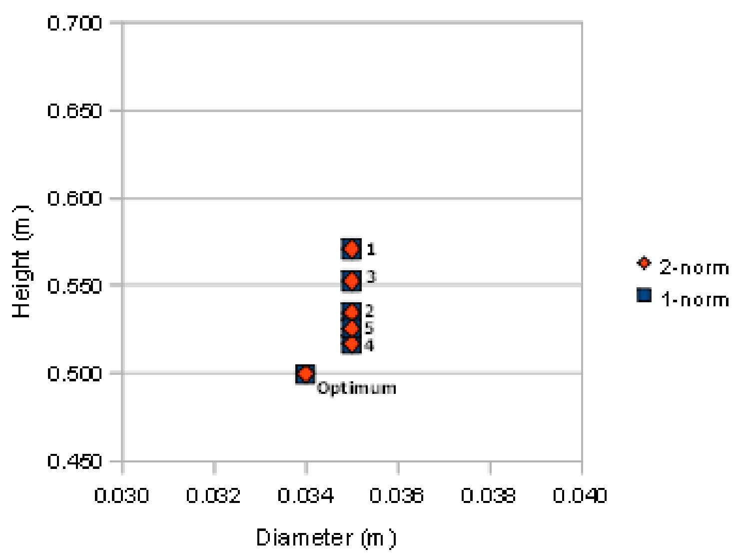


Figure 3.7 Alternatives Using Minimum Approach with 10% Performance Variability

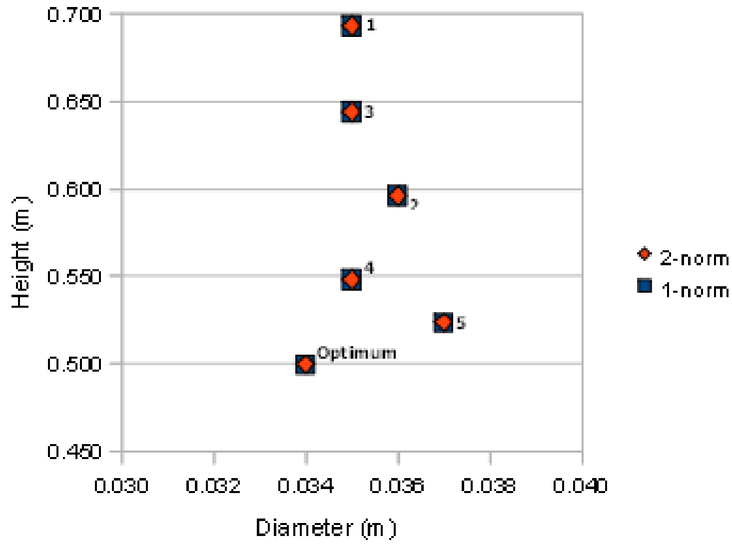


Figure 3.8 Alternatives Using Minimum Approach with 20% Performance Variability

Overall the results of this problem indicate that the tripod has greater design freedom along the height axis than along the diameter axis. This suggests that in order to maintain acceptable performance levels a designer should limit changes in the tripod design to only height changes. Additionally, it was noticed that both the 1-norm and the 2-norm metric found the same alternatives every time. This was not expected, but could be attributed to a lack of design freedom within this problem. In the next section, a multiobjective optimization problem is explored to assess the effectiveness of these two metrics at generating unique alternatives when more opportunities for change are present.

3.3.2 Vibrating Platform

In the previous section, a single objective case study was explored. This case study provided an overview of the MGA process. It also allowed for a modification to the MGA approach to be performed. This modification changed the way distance was calculated for multiple alternatives so that the alternatives identified would be more uniformly distributed throughout the design space. This section will test this modification on a multiobjective problem and see if both 1-norm and 2-norm distance continue to identify the same alternatives. Additionally, the effects of scaling techniques will be investigated as discussed in section 3.2.4. The nonlinear multiobjective engineering design problem chosen is the design of a vibrating platform [46]. The problem is composed of a vibrating motor sitting atop a platform comprised of three sandwiched materials resting on rollers. A representation of the vibrating platform is shown in Figure 3.9.

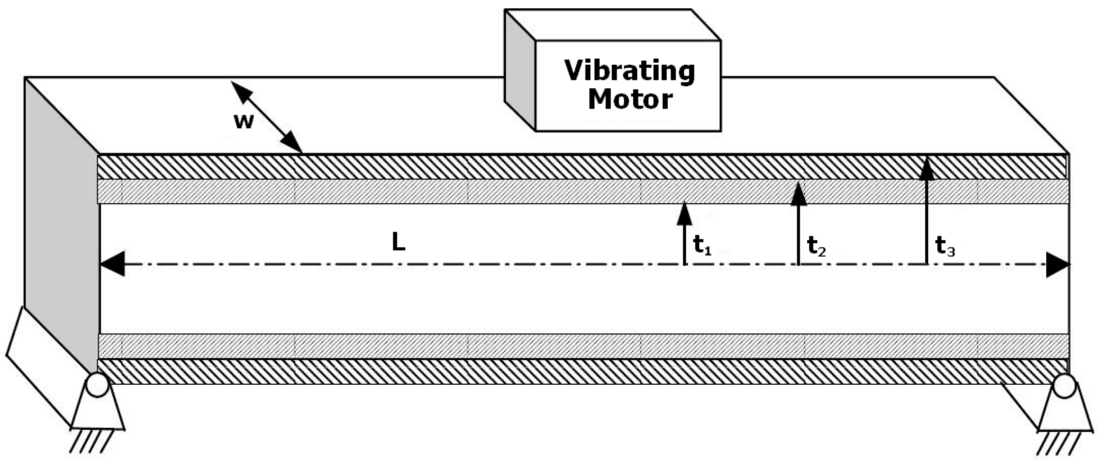


Figure 3.9 Vibrating Platform

The objectives of this problem are to maximize natural frequency and minimize cost. It is modeled by six design variables representing three thickness values (t_1 through t_3), the length of the beam (L), the width of the beam (w), and the order in which the materials are arranged (m). The material order variable (m) ensures that all three of the available materials, shown in Table 3.1, are used. Thus, the possible combinations of three materials leads to six discrete options, {A,B,C}, {A,C,B}, {B,A,C}, {B,C,A}, {C,A,B}, and {C,B,A}. This variable, when combined with the other five, yield a mixed-integer problem. Additionally, constraints maintain the thickness of the layers within a specified range and limit the mass of the platform. A formulation of the problem can be seen in Equation 3.7.

Table 3.1 Vibrating Platform Material Properties

m_i (*material*)	$\rho_i\left(\frac{kg}{m^3}\right)$	$E_i\left(\frac{N}{m^2}\right)$	$C_i\left(\frac{\$}{m^3}\right)$
A	100	1.6×10^9	500
B	2,770	70×10^9	1,500
C	7,780	200×10^9	800

$$
\begin{aligned}
&\text{Max: } f_1 = \left(\frac{\pi}{2L}\right)\left(\frac{EI}{\mu}\right)^{0.5} \\
&\text{Min: } f_2 = 2w[c_1 t_1 + c_2(t_2 - t_1) + c_3(t_3 - t_2)] \\
&\text{S.T. } \mu L - 2800 \leq 0 \\
&t_2 - t_1 - 0.15 \leq 0 \\
&t_3 - t_2 - 0.01 \leq 0 \\
&0.05\text{m} \leq t_1 \leq 0.5\text{m} \\
&0.2\text{m} \leq t_2 \leq 0.5\text{m} \\
&0.2\text{m} \leq t_3 \leq 0.5\text{m} \\
&0.35\text{m} \leq w \leq 0.5\text{m} \\
&3\text{m} \leq L \leq 6\text{m} \\
&EI = \left(\frac{2w}{3}\right)[E_1 t_1^3 + E_2(t_2^3 - t_1^3) + E_3(t_3^3 - t_2^3)] \\
&\mu = 2w[\rho_1 t_1 + \rho_2(t_2 - t_1) + \rho_3(t_3 - t_2)]
\end{aligned}
\tag{3.7}
$$

Unlike the single objective example, the solution to the problem is not provided. Therefore it was identified using a grid search. The resulting Pareto frontier can be seen in Figure 3.10. The grid search discretization for each design variable was 0.005 m for t_1 through t_3, 0.05 m for w, and 0.1 m for L. The material order variable (*m*) was already discretized via integer encoding. This discretization yielded a population of approximately 1.1 million feasible unique designs. Normally, a grid search would not be used to identify a Pareto frontier as it is computationally inefficient. However, in this case the grid search was designated to be used as a search technique as it provides a repeatable search technique for comparing different settings. Therefore it was actually more efficient to go ahead and use these results to create the Pareto frontier.

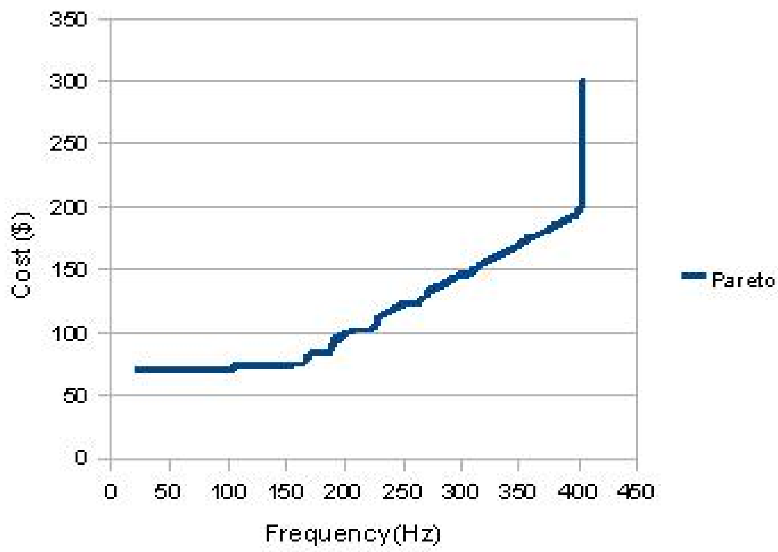


Figure 3.10 Vibrating Platform Pareto Frontier

As mentioned in section 3.2, there is not a single optimal point that serves as a convenient point of interest. Instead, three designs from the Pareto region are chosen: one at the upper knee, one in the middle, and one at the lower knee. The values for these three designs are shown in Table 3.2. These three designs represent possible solutions that a designer might choose based upon which performance objective they cared most about, cost or natural frequency.

Table 3.2 Vibrating Platform Points of Interest

	Design Variables						Performance Values	
Region	t_1	t_2	t_3	**w**	**L**	**m**	f_1	f_2
Upper	0.47	0.50	0.51	0.35	3	2	401.29	198.8
Middle	0.33	0.35	0.35	0.35	3	2	275.11	134.75
Lower	0.20	0.21	0.21	0.35	3	1	153.02	72.8

Having identified the Pareto frontier and the corresponding points of interest, the performance freedom must be defined. Much like the single objective problem, an exploratory 10% variance for both objective functions was selected. The resulting feasible performance regions are shown in Figure 3.11, with 577 designs in the upper region, 713 designs in the middle region, and 65 designs in the lower region. Immediately, it is noticed that using a percentage based performance freedom creates unequal amounts of performance freedom based on where in the performance space the point of interest is located. This variation in performance freedom can be visualized by looking at the colored regions in Figure 3.11.

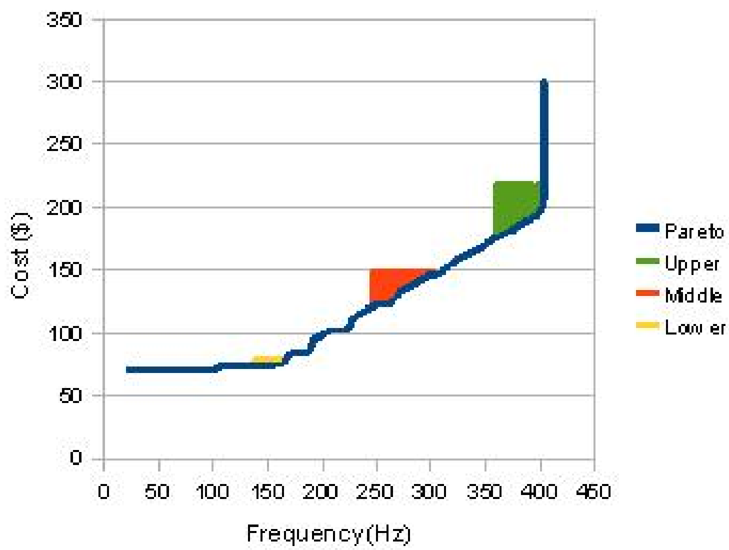


Figure 3.11 Performance Space with 10% Performance Variability in Both Objectives

As previously mentioned, the distance metrics chosen are the 1-norm and 2-norm given by Equation 3.1 and Equation 3.2, respectively. Additionally, it was found that a different technique for identifying multiple alternatives provided a more uniform representation of the design space. This minimum distance approach from section 3.3.1 will be adopted here. However, new challenges arising from the problem formulation of the vibrating motor were encountered. These challenges were not encountered in the single objective problem as it was not mixed-integer. The problem itself was caused by the encoding of material choice as an integer variable. For example, material order choice 6 appeared to be five units away from material order choice 1 as the difference between 6 and 1 is five. To remedy this situation, a hamming distance was used to calculate the difference between the design under evaluation and the alternative of interest.

The hamming distance metric compares each bit in the string of the current design with the bit in the corresponding alternative. If they are different, it increases the amount of change by 1. For instance, if the design under evaluation had a sequence of {A,C,B} and the alternative of interest had a material sequence of {A,B,C}, the amount of change would be two, since the second and third bits in the string changed. With this change in place, the results were much more intuitive.

3.3.2.1 Percent Difference Scaling

The first group of alternatives explored the results of the upper region for both the 1-norm and 2-norm metrics using percent difference scaling (Figure 3.12). Unfortunately, this scaling method did not work for the material order variable (*m*) as all percent changes equaled 200%. This can be attributed to the ‘design variable value’ in Equation 3.3 equaling 0. As previously mentioned, if either of the variables in a percent difference calculation equal zero the result will always equal 2. To remedy this, the material order variable was scaled by dividing the actual change by the the maximum possible change, which in this problem was three. The results are displayed using a radar chart with the axes on a $\log_{100}$ scale. The values on each axis represent the percent difference in each variable from the original point of interest in the upper region. It is interesting to note four unique alternatives are identified by each metric. Both metrics found one common pair of alternatives (1-norm #2 and 2-norm #3).

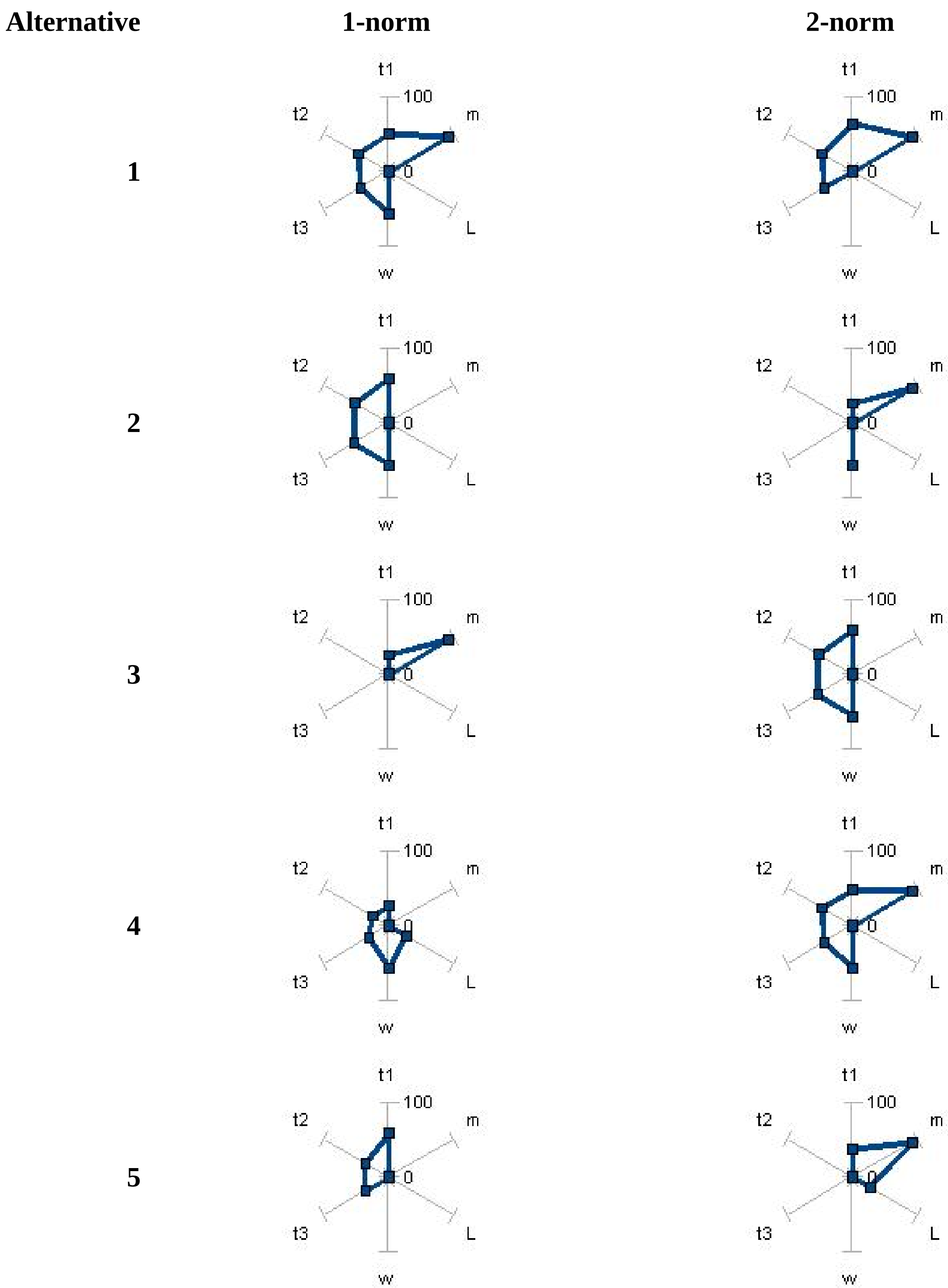


Figure 3.12 Upper Region Alternatives Using Percent Difference with 10% Performance Variability in Both Objectives

3.3.2.2 Bound Scaling

The second group of alternatives explored the results of the upper region for both the 1-norm and 2-norm metrics using a bound scaling technique (Figure 3.13). This time the scaling of the material order variable (*m*) could be performed in the same manner as the rest of the variables as difference between variables is divided by the maximum variable range instead of an average. Essentially, this is the same technique use to scale *m* for the group of percent difference alternatives. Again, the results are displayed using a radar chart with the axes on a $\log_{100}$ scale. The most interesting result from this group of alternatives was that both the 1-norm and the 2-norm identified the same first three alternative designs. Further, these three designs are the same as the first three designs found using the 1-norm approach with percent difference scaling.

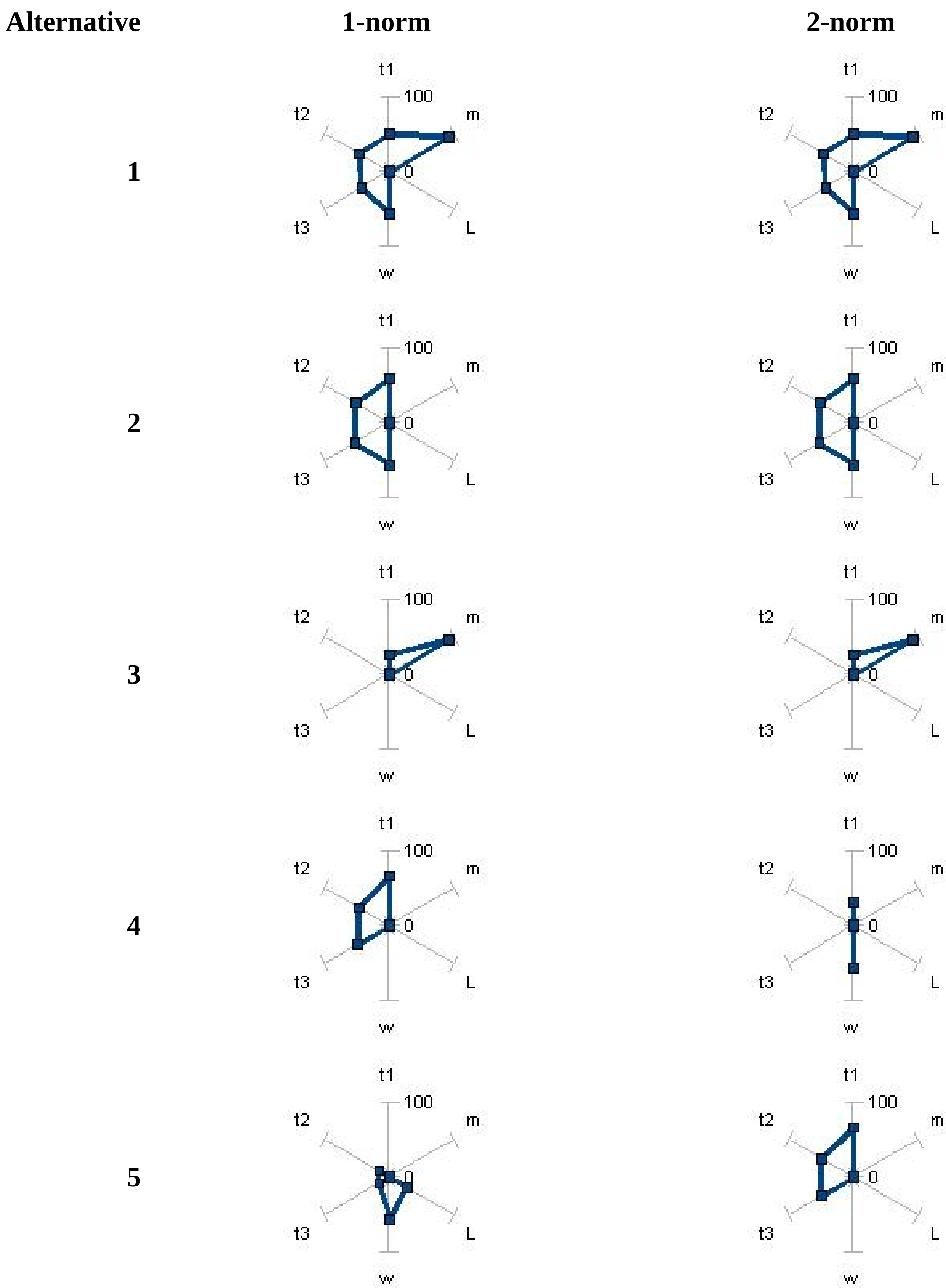


Figure 3.13 Upper Region Alternatives Using Bound Scaling with 10% Performance Variability in Both Objectives

3.3.2.3 Max Step Scaling

The third group of alternatives looked at the results of the upper region for both the 1-norm and 2-norm metrics using the max step scaling technique (Figure 3.14). The advantage of this approach is that the alternatives generated should leverage the maximum possible move for the design space. However, this approach is only practical for population based search techniques as these make it computationally inexpensive to find the range of each variable. Additionally, this approach was able to scale the material order variable (m) in the same manner as the other variables. Again, the results are displayed using a radar chart with the axes on a $\log_{100}$ scale. The most interesting result from this group of alternatives was that the 1-norm again found the same first alternative as the previous two groups. This gives some support to the idea that the 1-norm metric induces less bias into the alternative selection process, regardless of the scaling technique chosen. The overall results from this group of alternatives show that both the 1-norm and 2-norm metrics each found three unique alternatives. The two common alternatives were 1-norm #2 / 2-norm #3 and 1-norm #5 / 2-norm #4.

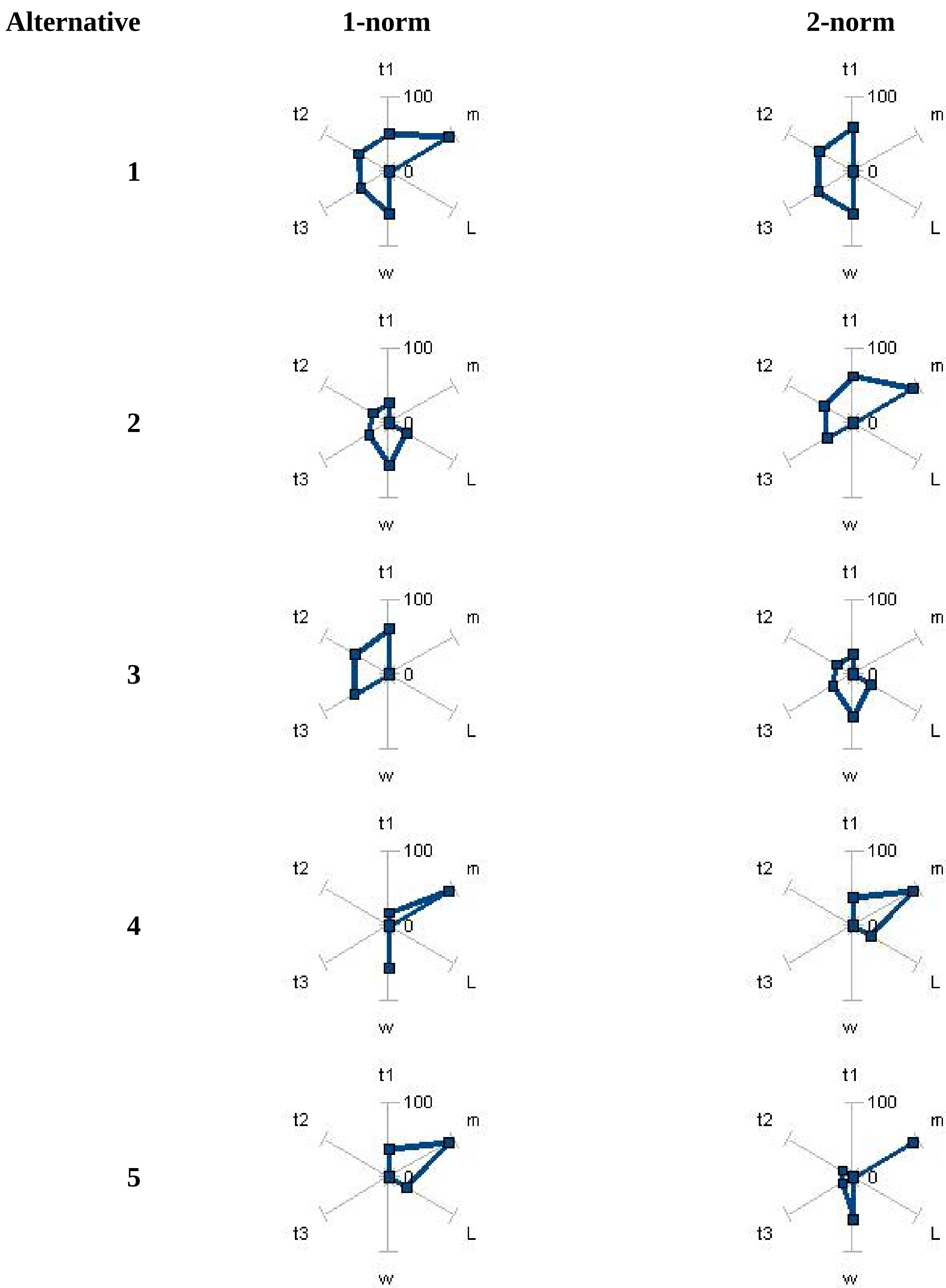
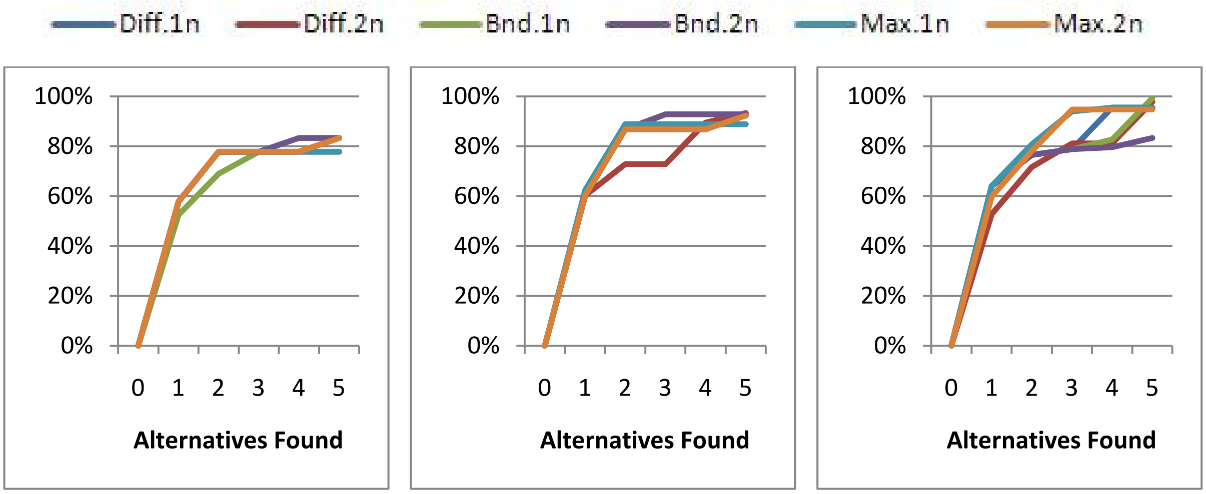


Figure 3.14 Upper Region Alternatives Using Max Step Scaling with 10% Performance Variability in Both Objectives

3.3.2.4 Feasible Design Space Capture Rate

The remaining two points of interest were tested in a similar manner. The results showed that there was not a single common alternative amongst all six combinations of distance metric and scaling technique for either the upper or middle region. However, there was a single common design within the lower region, which can likely be attributed to their being far fewer designs to choose from. Unfortunately, the individual designs themselves did little to suggest any trends that may be occurring. The only major trend noticed was that the alternatives identified by the 1-norm distance metric seemed to be less influenced by the scaling technique chosen. Therefore a different visualization was created to see if any trends could be found. As seen in Figure 3.15, the percent of feasible design space captured by each of the six combinations of distance metric and scaling technique. Within it, the x-axis represents the number of alternatives generated and the y-axis represents the percent of the feasible design space that was captured by the alternatives. The percent of feasible design space was calculated by first calculating the maximum variable range of the feasible design population. Then the variable change from the point of interest to each alternative was calculated. Each of these variables ranges was then divided by the maximum variable range of the feasible design population to create a vector representing the percent of each variable captured. Finally, this vector was averaged to create the percent of feasible design space measurement. Essentially, this measurement is the average percentage of captured variable range. The results from Figure 3.15 show that the feasible design space was captured quickly in all three alternative groupings. In fact, all but one case was able to capture 80% of the feasible space by the time the third alternative was identified. Additionally, Figure 3.15 shows that, of the three scaling techniques, max step was the most consistent at achieving high performance. In this case, performance was gauged by how quickly design space was captured.



Lower Region **Middle Region** **Upper Region**

Figure 3.15 Feasible Design Space Capture Rate

3.3.2.5 Non-Equal Performance Freedom

After seeing that the differences between the distance metrics and scaling techniques were minimal, an additional study was proposed. This study looked at the how performance freedom affects the alternatives identified. To accomplish this, four unbalanced combinations of performance freedom were created. These combinations were: [0%,10%], [5%,10%], [10%,5%], and [10%,0%]. Here, the first number denotes the allowed variation in natural frequency and the second number denotes the allowed variation in cost. As expected, reducing the performance freedom caused an increased number of common designs found. Recall, that previously only the lower region found a single common design. An example of this increased alternative commonality can be seen within the [10%,0%] performance freedom of the lower region. Within this case, all six combinations of distance metric and scaling technique generated the same five alternatives. In the upper region, common designs between the six combinations occurred when the performance freedom of natural frequency was reduced to 5% or 0%. This result is interesting as the upper region has the smallest tradeoffs in natural frequency on the Pareto frontier. Additionally, the results of this study support the notion that as the number of feasible designs increase, fewer common designs will be found by different combinations of distance metrics and scaling techniques.

3.4 Results

This chapter focused primarily on identifying the effects of different distance metrics and scaling techniques on nonlinear multiobjective engineering design problem. Additionally, it identified a few key changes that needed to be made so that MGA was able to find alternatives correctly.

The distance metrics tested included 1-norm and 2-norm distances. Results from section 3.3.1 showed that it is possible for each distance metric to find the same alternatives. However, this was in a single objective problem with only two variables. In the multiobjective problem the 1-norm and 2-norm distance were able to find some of the same alternatives. However, no trends as to why or when they would find the same alternative were discovered. A promising discovery within the distance metrics came from the 1-norm distance. In section 3.3.2 it showed the ability to find the same initial alternative regardless of the scaling used. This seems to support the idea that alternatives found using a 1-norm distance metric are less affected by scaling than alternatives found using a 2-norm distance metric. This apparent robustness is a desirable trait and if it is able to be more thoroughly proven, 1-norm distance provides a good starting distance metric candidate.

The scaling techniques tested included percent difference scaling, bound scaling, and max step scaling. Of these three, max step scaling showed the most consistent performance capabilities. However, the performance of all three scaling techniques was very close, especially for the first few alternatives identified. It was also shown that for 10% performance freedom all but 1 out of 18 alternative searches captured close to 80% of the available design variable range. It was also demonstrated that percent difference scaling is unacceptable for use when design variables approach a value of zero.

Five combinations of performance freedom were tested. These included performance freedom combinations of [0%,10%], [5%,10%], [10%,10%], [10%,5%], and [10%,0%]. Again, the first number denotes the allowed variation in natural frequency and the second number denotes the allowed variation in cost. The results of this study gave an indication of correlation between the number of feasible designs and the number of common designs discovered. More precisely, as the number of feasible designs increase more common designs were found using different MGA settings. Additionally, there seems to be an increased effect on the tails of the Pareto frontier due to changes in performance freedom. This could possibly be attributed to the tails restricting the number of designs much more aggressively than the rest of the Pareto frontier.

The changes performed to MGA to ensure proper alternative identification in nonlinear multiobjective problems were surprisingly minimal. The first change was identified in section 3.3.1. This change modified how MGA dealt with distance to multiple alternatives. The previous approach called for the summation of the distance to each successive alternative. However, this causes the alternatives to cluster towards the extremes of the feasible design space. It was found that using only the distance to the closet alternative provided a much better method of calculating the distance to multiple alternatives. This "minimum" approach caused the alternatives to be much more evenly distributed in the design space.

The second change needed was due to a mixed-integer problem. It was found that the 1-norm and 2-norm distances were handling an integer choice variable improperly. To solve this, a hamming distance was used to calculate the effective difference between the strings that the integers represented. The hamming distance was then included in with the rest of the 1-norm or 2-norm distance calculation.

Finally, this chapter verified that MGA can be used to identify useable alternative designs in nonlinear multiobjective engineering design problems with minimal changes. However, to accomplish this a computationally expensive, but robust, grid search algorithm was used. Ideally, MGA would be able to be performed with much less computational overhead. The next chapter identifies search techniques that make MGA computationally viable.

CHAPTER 4: COMPUTATIONAL EFFICIENCY USING GRAVEYARD DATA

4.1 Motivation

Recall that multiobjective problems are likely to require higher computational cost than single objective problems due to the increased number of functions that have to be evaluated. Again this is stated as one of the major reasons for not using MGA on multiobjective problems. However, this chapter strives to overcome this hurdle by reusing the designs evaluated during the search for the Pareto frontier, i.e. graveyard data. It is hypothesized that using graveyard data can dramatically reduce the computational expense needed to find alternative designs. Additionally, this chapter will also measure the error of each search technique as compared to an ideal alternative design solution. This can be attributed to the observation that gains in one area typically come at the expense of another. In the following section the method used to measure this computational cost and error of each search technique is presented.

4.2 Method

In this section the method used to quantify the computational costs and measure the error for each search technique is discussed. The method, as seen in Figure 4.1, is applied to two different nonlinear multiobjective engineering design problems; the design of a two bar truss and an I-beam. These problems are chosen as there is existing design freedom information for them [50-51].

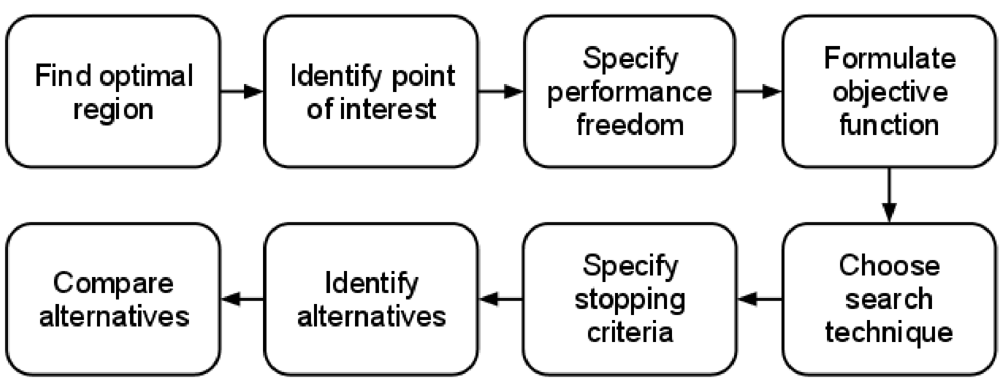


Figure 4.1 Research Question #4 Method Flow Chart

4.2.1 Find Optimal Region

The method starts with modeling a problem and finding the Pareto frontier using a multiobjective genetic algorithm (MOGA). Recall that MOGAs are part of the evolutionary algorithm family, which is already used to overcome modeling uncertainty (Section 2.2.1). Further, MOGAs tend to evaluate a large number of designs near the frontier. This increased density near the frontier is desirable for an alternative design search as there are a large number of design choices in a very small region. These make MOGAs a good candidate for creating graveyard data. Frontiers presented in this chapter were found using the gamultiobj() function in the Matlab Optimization Toolbox [59].

4.2.2 Identifying Point of Interest

As with the previous research question, three representative points of interest from the Pareto frontier will be investigated to eliminate any biasing that may occur in certain regions of the Pareto frontier. These points reside on the Pareto frontier at a weighted L2 norm of [25%, 75%], [50%, 50%], and [75%, 25%].

4.2.3 Specify Performance Freedom

As previously mentioned, multiple levels of performance freedom will be specified for this chapter. This is done to provide a deeper understanding of how each search technique performs with respect to performance freedom. The levels chosen for this work are from 1% to 100% of the performance bounds of the Pareto frontier in each objective. For the results presented in this thesis, the MGA approach is run with a step size increase of 1% in all objectives. At 100%, the search techniques are allowed to explore the design space and find alternatives that fall within the extremes of the Pareto front. An example of this extreme case is represented by the shaded maximum performance freedom region in Figure 4.2. For this example, there are two objectives (f_1 and f_2), a Pareto frontier (black dots), and a performance freedom region (gray area).

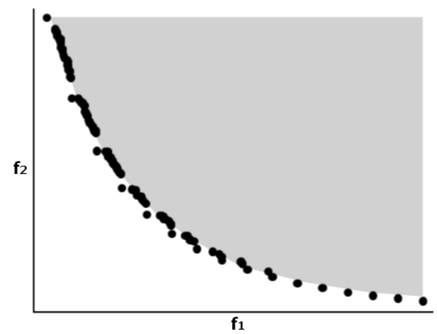


Figure 4.2 Maximum Performance Freedom

4.2.4 Formulate Objective Function

Recall that the goal of the objective function in this research is to maximize the distance in the design space while remaining within a specified performance threshold. To reduce the number of independent variables, only 2-norm distance and bound scaling will be used. The formulation for 2-norm distance was previously shown in Equation 3.2. While other distance metrics can be used [60], 2-norm distance is more commonly encountered within the vector-based math that is common in engineering design problems [61]. Additionally, it was shown in the previous chapter that when bound scaling is used there is little difference between 1-norm and 2-norm distance. Bound scaling is chosen as both test problems supply variables bounds. Additionally, bound scaling does a better job of scaling all the variables equally.

4.2.5 Choose Search Technique

The “search” techniques to be tested include a graveyard, Latin hypercube sampling, grid search, and a Nelder-Mead simplex method. These are chosen as they each bring different characteristics to the alternative search.

One source of differentiation between these search techniques, for example, is that the graveyard is highly dense near the Pareto frontier, while the grid search is more widely distributed throughout the performance space. A similarity between the grid search and Latin hypercube sampling is that there is a degree of assurance that the full range of each variable is represented in the design space. The other search techniques do not come with the same guarantee, as there are some governing rules that direct their search of the design space.

A distinguishing characteristic of the Nelder-Mead simplex method is that computational cost depends on the starting point and design variable interactions within the model. This is significant, as the other search techniques used in this thesis have a constant computational cost that can be calculated prior to performing the search.

To perform the grid search, small steps are taken along each variable axis in a sequential manner over the entire allowable range. The step sizes used in this paper are a 1% discretization for the two bar truss problem and 2% for the I-beam problem. Limiting factors of the step size chosen include computational time and available memory.

For the graveyard search, five independent runs of a multiobjective genetic algorithm were performed on each case study. The reason for creating multiple sets of graveyards comes from the randomness associated with a genetic algorithm. Each run of the genetic algorithm began with a random initial population. The rest of the settings were the defaults as per the gamultiobj algorithm within Matlab [59].

Two different sets of latin hypercube samples were created. The first set used a sample size equal to 100 times the number of design variables (LHS100), while the second set used 1,000 times the number of design variables (LHS1000). Both sets were had five independent samples performed for each case study. Again this was done to account for randomness within the process.

The last of the "search" techniques, the Nelder-Mead simplex method, was performed using the fminsearch algorithm within Matlab [62]. This technique was then subdivided into two different techniques, a direct search and a hybrid search.

The direct search uses only the point of interest as its starting point, while the hybrid search uses the best alternative from the corresponding graveyard as its starting point. The direct search is only run once as its starting point does not change and the results are repeatable. However, since there are five graveyards populations created, the hybrid search is performed five times, i.e. once for each graveyard. By combining information from the graveyard with the Nelder-Mead simplex method, it is hypothesized that computational cost and error will be decreased because of the improved starting location.

4.2.6 Specify Stopping Criteria

For both case studies in this chapter, MGA was used to find only the first alternative. Since, this chapter will be measuring error it is important to ensure that the data comes from independent samples. Each alternative is dependent on the alternatives and the point on interest that exist in the design space. However, since the first alternative depends only on the point of interest, which does not change, it can safely be used to compare the different search techniques.

4.2.7 Identify Alternatives

Groups of alternatives are found for each point of interest. This results in three different groups of alternatives for each case study. These groups can be further subdivided into six sub-groups. Each sub-group corresponds to one of the six scaling techniques.

4.2.8 Compare Alternatives

Each of the sub groups identified in the previous section is compared to the other sub-groups on the basis of error and computational cost. In this chapter, error is measured by finding the difference between two distance measurements. The two measurements considered are the distance to an ideal alternative design and the distance to the current alternative design. The ideal alternative design, which this thesis will refer to as the "best known" alternative design, is chosen from the population of all the search techniques tried on a certain point of interest.

The decision to use distance as a measurement of error is grounded in the notion that two alternatives that are an equal distance away from the ideal point may have vastly different design variable values. However, since the objective of MGA is to find unique designs, there is no penalty added for how that uniqueness is achieved. Additionally, an alternative design having a superior distance measurement can theoretically encompass a large range of inferior alternative designs. A simplified example of this can be seen in Figure 4.3.

In Figure 4.3, two alternatives have been identified at different distances from the point of interest, denoted by an "x". If an arc is swept out at the distance corresponding to the superior alternative (i.e. the one with the larger distance) it is immediately noticed that the inferior alternative is contained within the arc. Further, any future alternative that lies on the arc would be considered equivalent. Moreover, a single distance value is far easier to comprehend than an array of individual variation measurements.

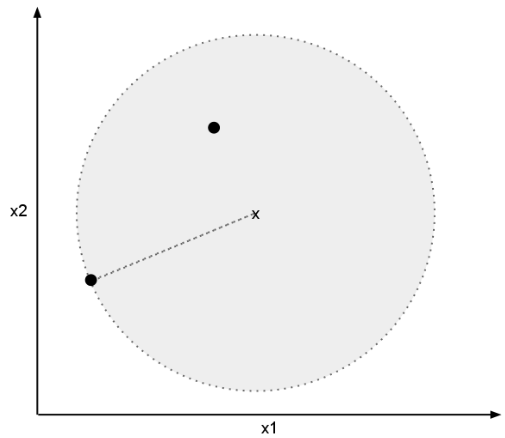


Figure 4.3 Equivalency of Distance Measurements

After quantifying the error for a point of interest, a statistical analysis is performed. First, an unbalanced one way analysis of variance (ANOVA) is performed on the error over the entire performance freedom range. The reason for the ANOVA being unbalanced comes from the fact that some search techniques have five replications (graveyard, Latin hypercube, Hybrid) while others only have one (grid search, Nelder-Mead). The ANOVA is used to test the hypothesis that the error for all of the search techniques is equal. This is similar to a T-test, except that a T-test is limited to comparing only two search techniques at a time. If the hypothesis is false, the 95% confidence interval for each search technique's error is calculated and plotted for comparison.

To calculate the computational cost of each sub-group, which corresponds to a search technique, the number of objective function evaluations is used. Function evaluations were chosen over evaluation time as it is not biased by the computer hardware used. For the two case study problems presented in this chapter, both problems have two objectives. Thus, the computational cost will always be an even number.

After determining all computational costs for a given point of interest, 95% confidence intervals for the computational cost of the direct search and hybrid are created. The rest of the search techniques have a confidence interval of zero due to them not varying in size.

In the next section, the approach outlined above will be applied to two different case studies. The first is the design of a two bar truss, which is known to have high design freedom near the frontier [50-51]. Regions with high design freedom are characterized by having lots of designs that possess similar performance characteristics. The second case study is the design of an I-beam. Previous work has shown that this problem lacks design freedom near the frontier [50-51].

4.3 Case Studies

In this section, the error and computational cost for six (graveyard, grid search, 2 Latin hypercube scenarios, Nelder-Mead, and Hybrid Nelder-Mead) search techniques are calculated and discussed for two different case studies. The first case study involves the design of a two bar truss. The second is the design of an I-beam. Within both problems it is assumed that the Pareto frontier has already been found using a multiobjective genetic algorithm.

4.3.1 Two Bar Truss

The Two-Bar Truss problem was chosen as a case study because it was previously shown to have good design freedom near part of the frontier [50-51] as seen by the contours in Figure 4.5. Problem formulation is from [46], and the problem consists of three design variables, two constraints, and two objective functions, as shown in Equation 4.1. The three design variables are the cross-sectional area of link AC (x_1), the cross sectional area of link BC (x_2), and the vertical position of the load (y) as denoted in Figure 4.4. The two objectives for this problem are minimize the stress in link AC (f_1) and minimize the amount of material used (f_2). The stress in link BC (g_1) is used as a constraint.

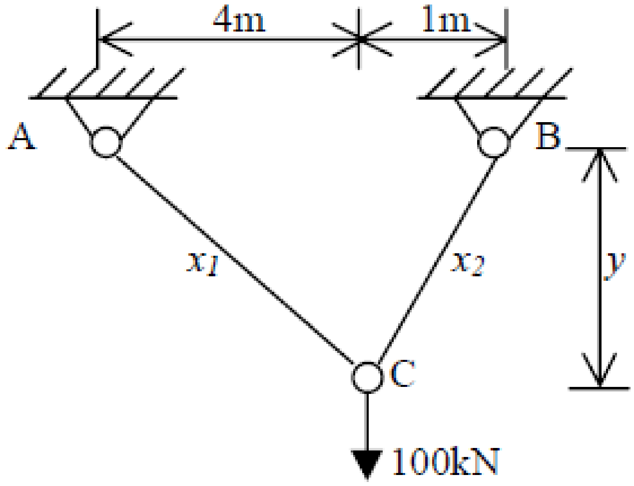


Figure 4.4 Diagram of Two Bar Truss

$$
\begin{aligned}
Min: f_1 &= \frac{20\sqrt{16+y^2}}{x_1 y} \\
f_2 &= x_1\sqrt{16+y^2} + x_2\sqrt{1+y^2} \\
S.T.\, g_1 &= \frac{80\sqrt{1+y^2}}{x_2 y} \leq 100{,}000 \\
f_1 &\leq 0.1 \\
f_2 &\leq 100{,}000 \\
Where: 0 &< x_1 \leq 0.0243 \\
0 &< x_2 \leq 0.707 \\
1 &\leq y \leq 3
\end{aligned}
\tag{4.1}
$$

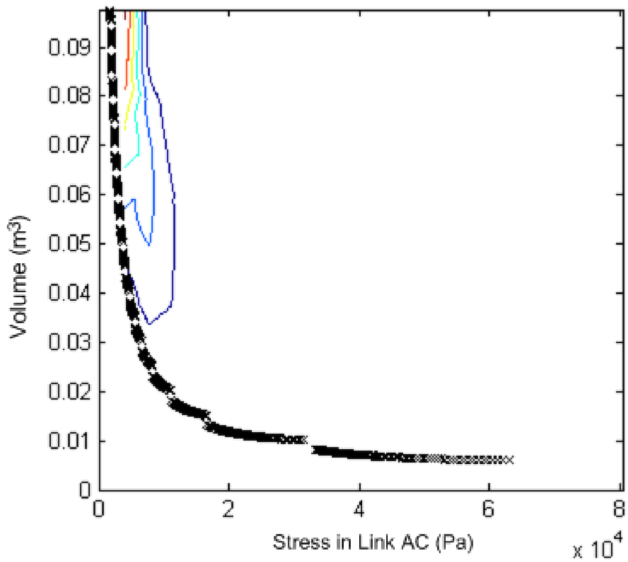


Figure 4.5 Two Bar Truss Design Freedom

The Pareto frontier for this problem is shown in Figure 4.6. From this frontier, three points of interest are chosen. The first point of interest is nearest to a known region of design freedom. The design and performance space information for all three points are interest are shown in Table 4.1, and their locations on the Pareto frontier are indicated by circles in Figure 4.6. Also, the maximum range of the performance freedom is calculated from the frontier. The maximum performance freedom range is 97.5 kPa along the horizontal f_1 axis, and 0.1 m^3 along the vertical f_2 axis.

Table 4.1 Two Bar Truss Points of Interest

	Design Variables			Performance Values	
Point of Interest	$x_1(m^2)$	$x_2(m^2)$	$y(m)$	$f_1(Pa)$	$f_2(m^3)$
#1	4.9e-3	0.99e-3	2.66	7.4e3	26.2e-3
#2	2.9e-3	0.86e-3	2.78	12.1e3	16.6e-3
#3	1.9e-3	0.98e-3	2.46	20.3e3	11.4e-3

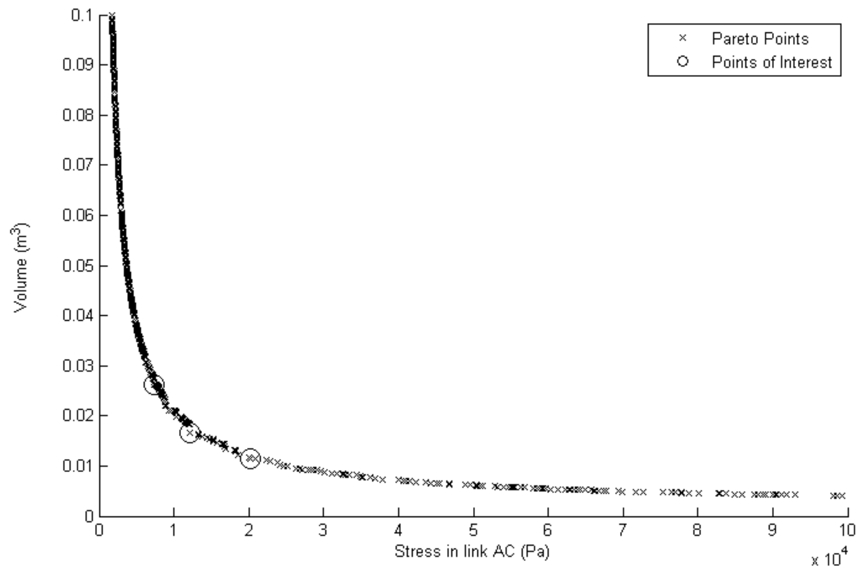


Figure 4.6 Two Bar Truss Pareto Frontier

4.3.1.1 Distance vs. Performance Freedom

Three images that demonstrate the correlation between search technique effectiveness and amount of performance freedom are shown in Figures 4.7-4.9. Each figure has seven curves. There is one that represents the average of each search technique. Additionally, there is a curve that depicts the design space distance from the point of interest to the ideal alternative for each amount of performance freedom. From the three figures it is noticed that the distance to the ideal alternative design increases with performance freedom. It is also evident that the error – or the inability to find the ideal alternative – typically increases with performance freedom. Of the seven curves, the LHS100 curve is consistently the worst performing in all three figures. This is likely a product of it having the smallest number of feasible designs in its population. Meanwhile, the grid search and both versions of the Nelder-Mead perform very well as they all display very little error from the best known case. The graveyard's performance is just as good as the Nelder-Mead searches and the grid search up to 10% performance freedom. After that point however, it displays an increasing amount of error from the best known alternative.

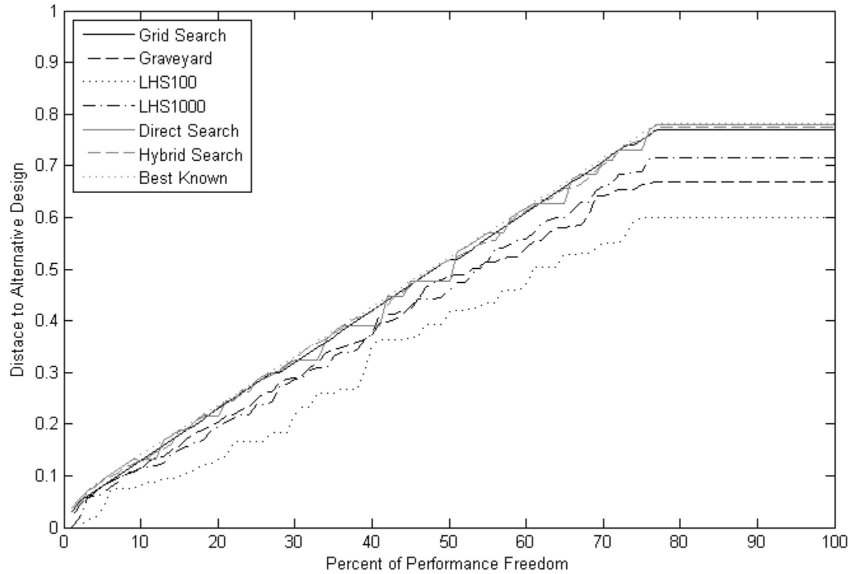


Figure 4.7 Distance vs Performance Freedom for Point of Interest #1

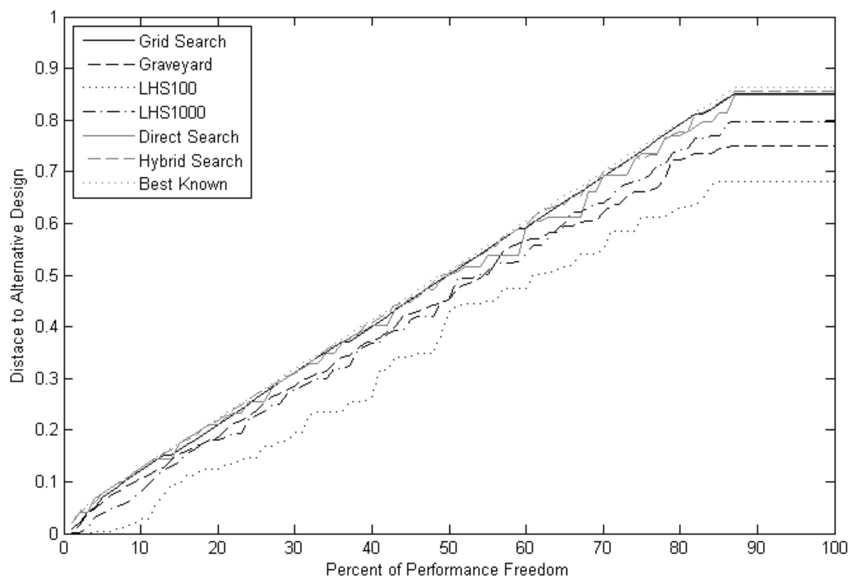


Figure 4.8 Distance vs Performance Freedom for Point of Interest #2

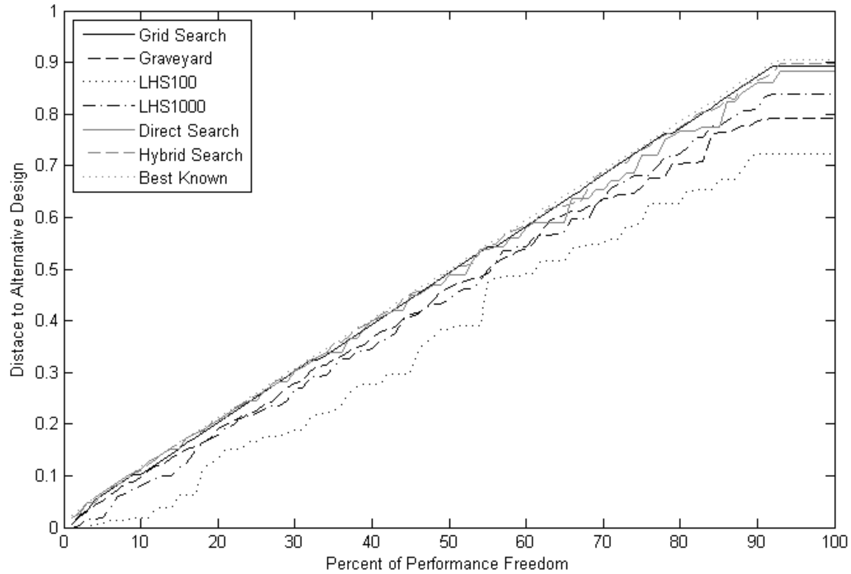


Figure 4.9 Distance vs Performance Freedom for Point of Interest #3

4.3.1.2 Confidence Intervals on Error

Running an ANOVA on the results from the previous section show that the resulting p value for all three points of interest is zero. This suggests that there is a 0% chance that the error in all six search techniques is equivalent. To explore the differences between the three approaches, a figure is created for each point of interest. These figures, Figures 4.10-4.12, compare each search technique's confidence interval with respect to error. These figures show that the grid search, the direct search, and the hybrid search consistently find an alternative with very little error, if any. On average, the direct search is the worst of these three search techniques. Another interesting discovery is the large confidence intervals found in both the grid search and the direct search. Since this result occurs in only these two search techniques, it is hypothesized that this is likely a product of them using 1/5 the sample size compared to the other search techniques. Performance of the graveyard search is shown to have error statistically equivalent to the LHS1000 search technique for all three points of interest. Further, the error is on the same order of magnitude as the grid and direct searches.

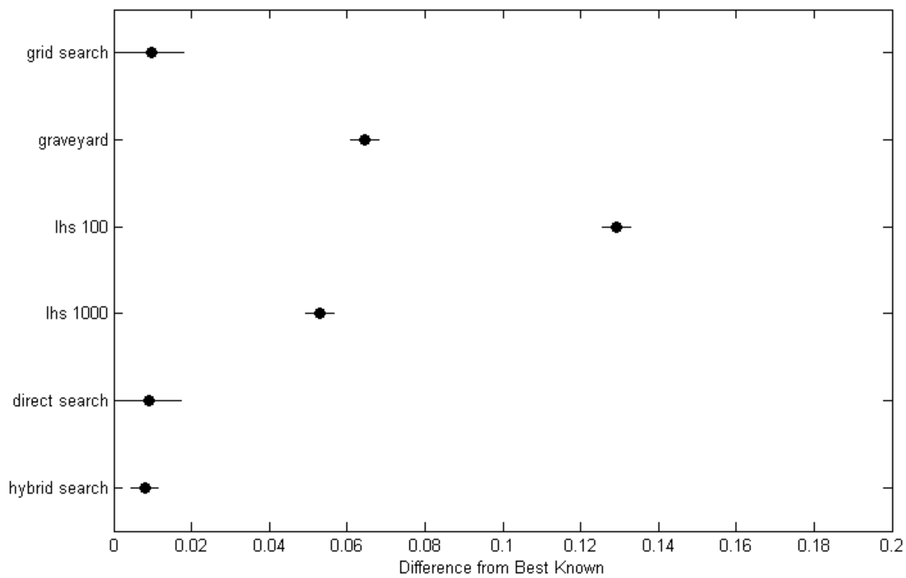


Figure 4.10 Error for Point of Interest #1

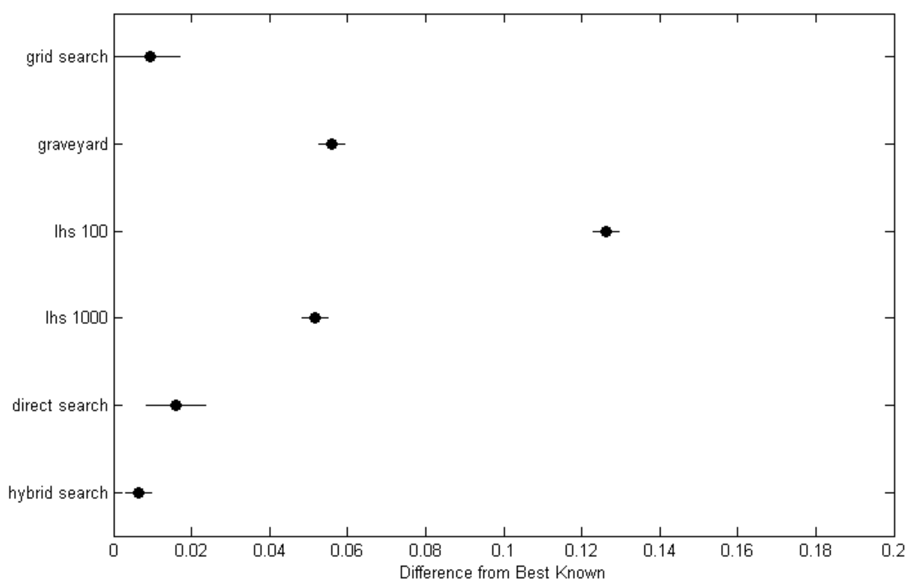


Figure 4.11 Error for Point of Interest #2

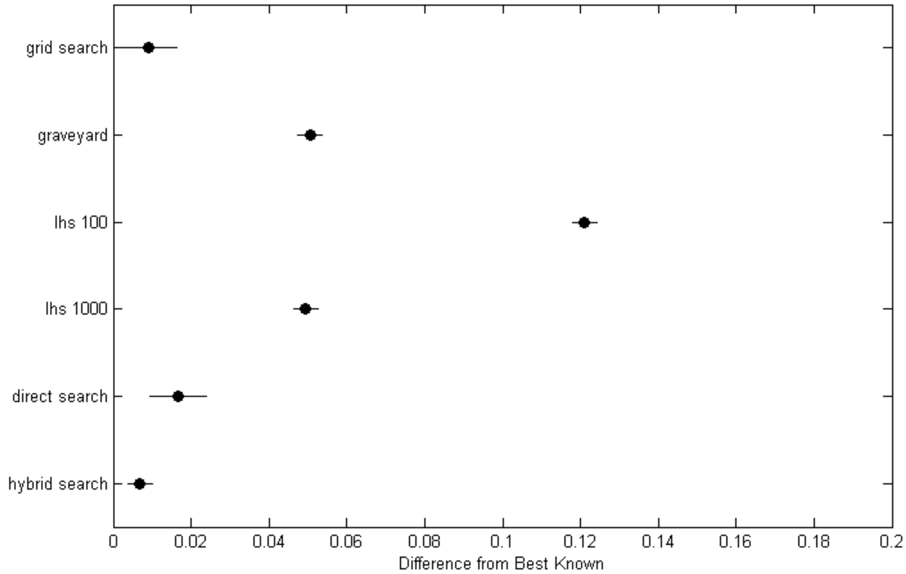


Figure 4.12 Error for Point of Interest #3

4.3.1.3 Computational Cost vs. Error

To visualize the trade-offs in computational cost and error, bubble plots of the six search techniques are shown in Figures 4.13-4.15. These plots show that four of the six search techniques exist on, or close to, the front of the trade-off curve in at least one of the figures. The two search techniques that use the graveyard (graveyard and hybrid) are part of this group. In fact, they are the only two search techniques not dominated by another search technique in at least one of the three figures. A dominated search technique is one that has both worse accuracy and worse computational cost than a different search technique. For example, in Figure 4.13, LHS100 has a higher computational cost and higher error than the graveyard. Thus, the LHS100 is dominated by the graveyard.

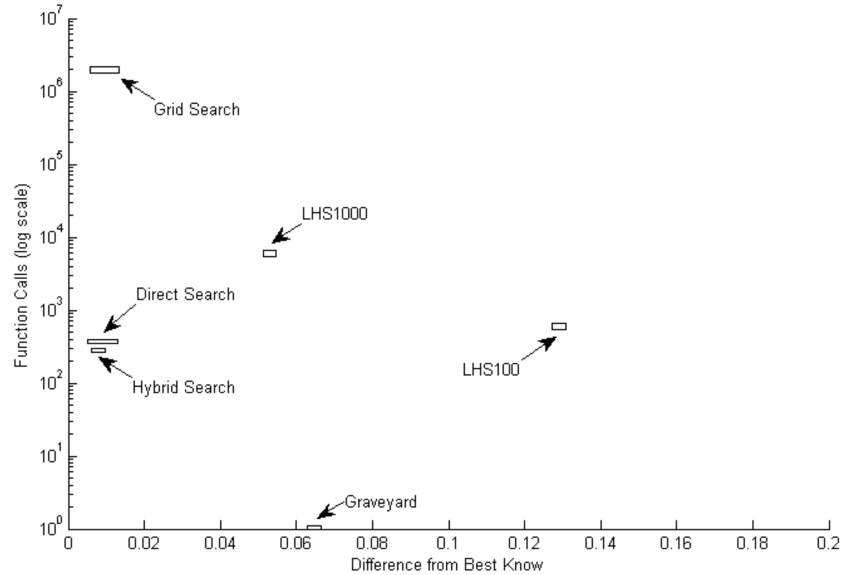


Figure 4.13 Tradeoffs for Point of Interest #1

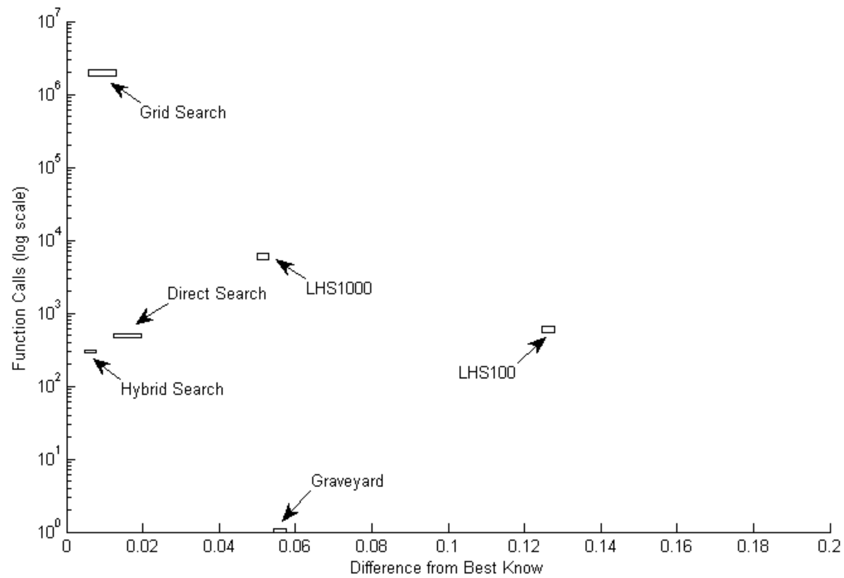


Figure 4.14 Tradeoffs for Point of Interest #1

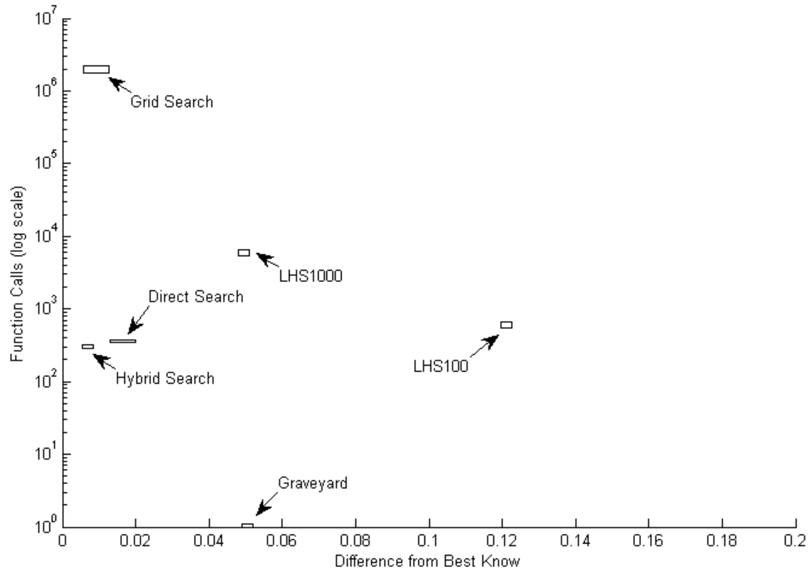


Figure 4.15 Tradeoffs for Point of Interest #1

For this problem, the findings indicate that using the graveyard in a problem with high design freedom is a good way to increase the performance of the MGA approach. In the next section, a multiobjective I-Beam case study problem is introduced to investigate the effects of having design freedom farther away from the Pareto frontier.

4.3.2 I-Beam

The I-beam problem is chosen as it has been shown to have poor design freedom near the Pareto frontier [50-51]. This problem is adapted from Hacker [63], and has four design variables and two objective functions, as shown in Equation 4.2. The physical representation of the four design variables are shown in Figure 4.16. The two objectives for this problem are: minimize the cross sectional area (f_1) and minimize the vertical deflection of the beam (f_2). The stress in the beam (g_1) acts as a constraint.

Additionally, a small, temporary change was made to the upper bounds of the design variables during the Latin hypercube sampling to ensure enough feasible designs were found. Using the original bounds, the LHS100 was returning less than ten feasible designs. Reducing the upper bound of x_1 and x_2 by roughly 50% allowed the LHS100 to achieve sample sizes greater than ten.

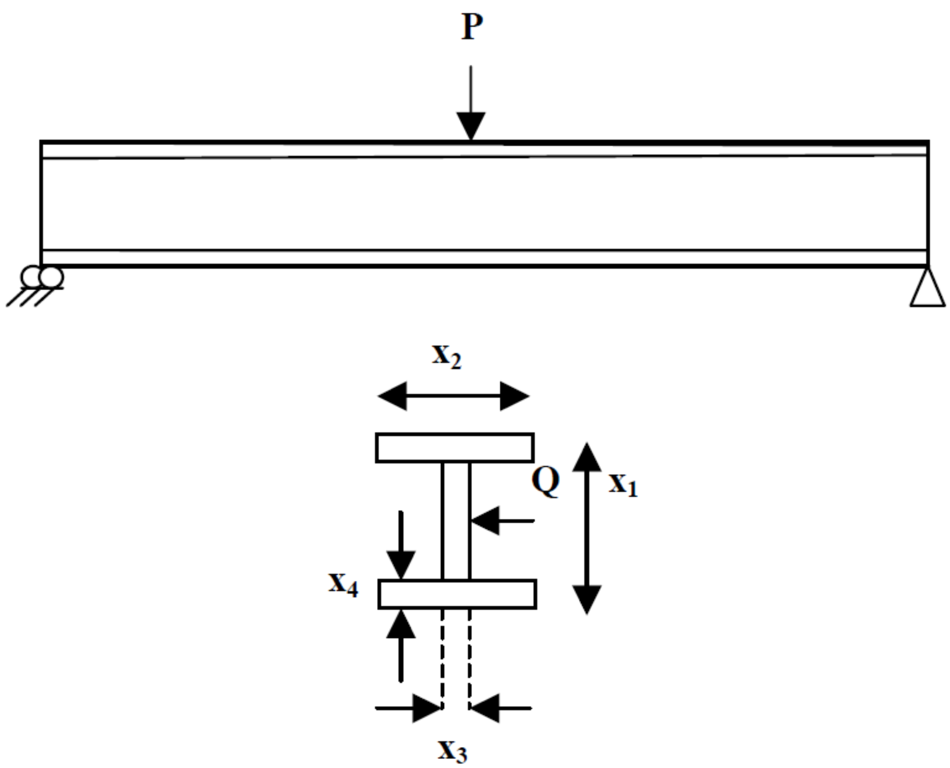


Figure 4.16 Cross Section of I-Beam

$$
\begin{aligned}
Min: f_1 &= 2x_2x_4 + x_3(x_1 - 2x_4)^3 \\
f_2 &= \frac{5000}{\frac{x_3}{12}(x_1 - 2x_4)^3 + \frac{x_2x_4^3}{6} + 2x_2x_4\left(\frac{x_1 - x_4}{2}\right)^2} \\
S.T.\, g_1 &= \frac{180e3x_1}{x_3(x_1 - 2x_4)^3 + 2x_2x_4\left(4x_4^2 + 3x_1(x_1 - 2x_4)\right)} + \frac{15e3x_2}{(x_1 - 2x_4)x_3^3 + 2x_4x_2^3} \\
&\le 16
\end{aligned} \qquad \textbf{(4.2)}
$$

$$
\begin{aligned}
\text{Where: } 10 &\le x_1 \le 80 \\
10 &\le x_2 \le 50 \\
0.9 &\le x_3, x_4 \le 5
\end{aligned}
$$

The Pareto frontier for this problem is shown in Figure 4.17, along with the three points of interest. The values of these locations in the design and performance space are shown in Table 4.2. It is interesting to note that all three points of interest have identical x_2, x_3, and x_4 values, but exist in entirely different regions of the performance space. Also, the maximum range of performance freedom is calculated from the frontier. The maximum performance freedom range is 0.19 cm^2 along the horizontal f_1 axis and 11.2 cm along the vertical f_2 axis.

Table 4.2 I-Beam Points of Interest

	Design Variables				Performance Values	
	$x_1(cm)$	$x_2(cm)$	$x_3(cm)$	$x_4(cm)$	$f_1(cm^2)$	$f_2(cm)$
#1	11.4	10	0.9	0.9	0.27	8.9
#2	17	10	0.9	0.9	0.32	3.5
#3	22.6	10	0.9	0.9	0.37	1.8

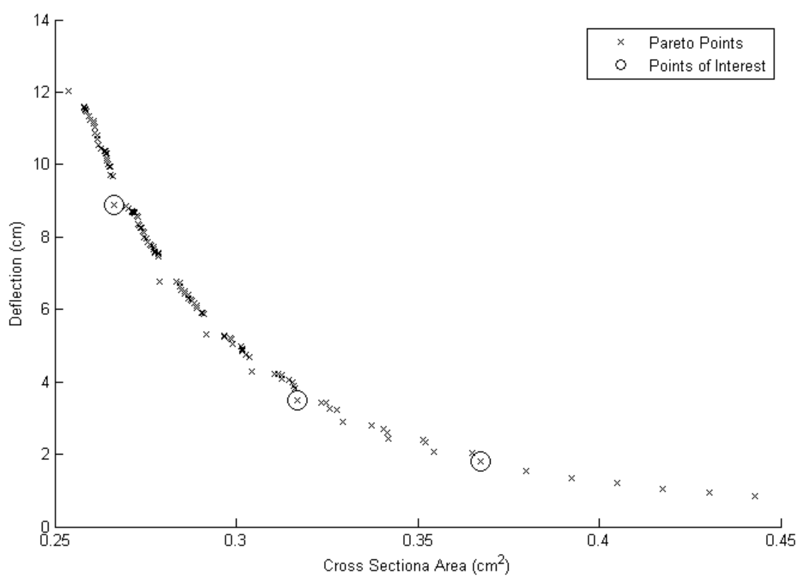


Figure 4.17 I-Beam Pareto Frontier

4.3.2.1 Distance vs. Performance Freedom

The correlation between the distance to the discovered alternatives and their respective performance freedom is shown in Figures 4.18-4.20. As before, each figure has seven curves: one for the average of each search technique and an additional curve for the best known alternative at each performance freedom. From these figures, it is noticed that distance increases with performance freedom for all search techniques. An interesting discovery is the reduction of error in Figure 4.20. This reduction in error is the result of no alternative design improvements found after the 50% performance freedom mark. Also in Figure 4.20, the direct search is shown to struggle finding good alternative designs. This result demonstrates the inability of the direct search to overcome the attraction of a locally good alternative design. A final discovery is that the error of the graveyard appears to increase dramatically after the 50 percent performance mark in Figures 4.18 and 4.19. However, closer inspection reveals that this is a side effect of the relatively small scale on the y-axis. The amount of error increase is comparable to the Two-Bar Truss case study.

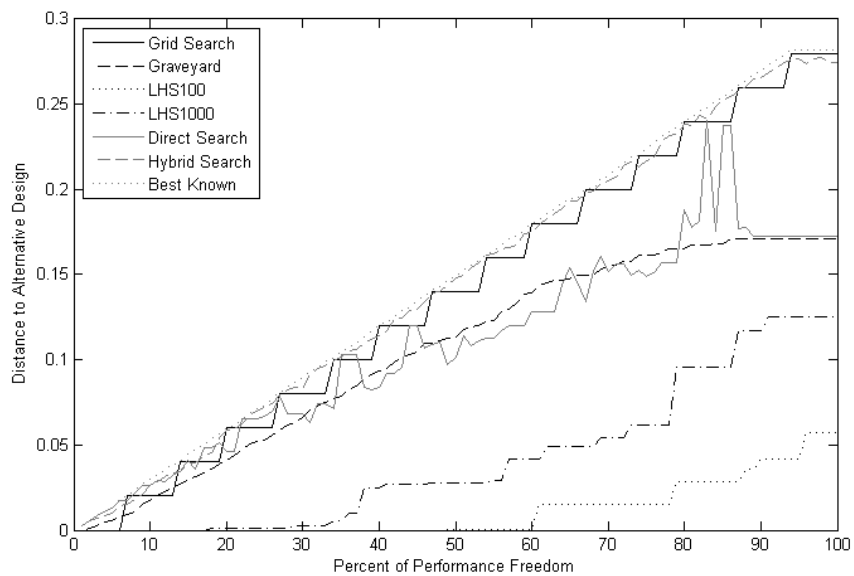


Figure 4.18 Distance vs Performance Freedom for Point of Interest #1

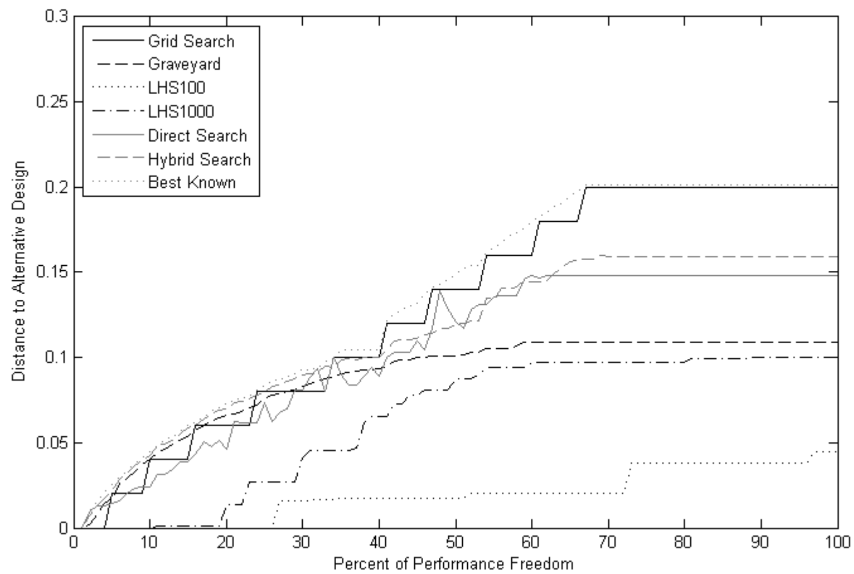


Figure 4.19 Distance vs Performance Freedom for Point of Interest #2

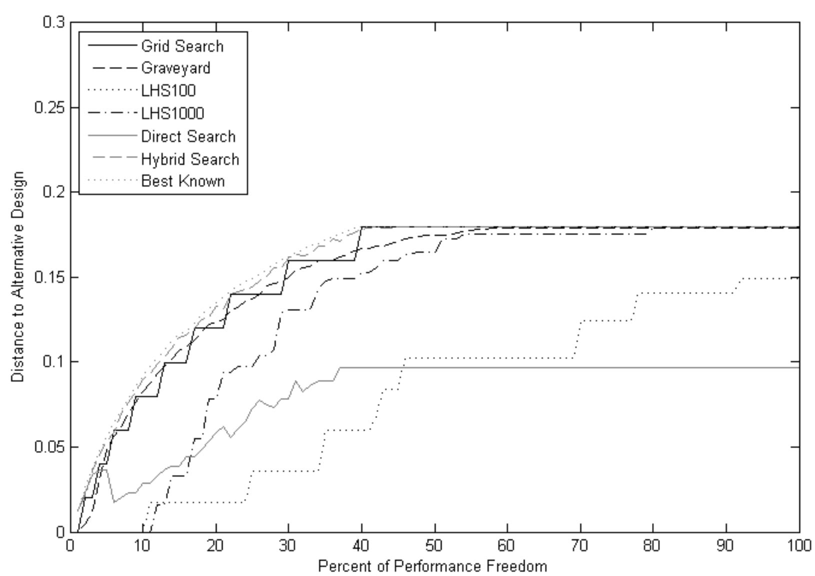


Figure 4.20 Distance vs Performance Freedom for Point of Interest #3

4.3.2.2 Confidence Intervals on Error

As in the previous case study, an ANOVA finds that the resulting p value for all three points of interest is zero. This suggests that there is a 0% chance that the error in all six search techniques is equivalent. Figures 4.21-4.23 compares the error for each search technique at each point of interest. For this case study problem, there is a much greater variation in the error of the search techniques.

The grid search is the only technique with an error consistently near zero. The graveyard and hybrid search are both as good as, and better than, the direct search, LHS100, and LHS1000 for all three points of interest. In Figure 4.23, the approach using the graveyard is shown to have very small error. This corresponds to the point of interest that struggled to identify better designs after the 50% performance freedom mark. This supports the notion that the graveyard has very low error corresponding to lower allowances of performance freedom. This idea is further reinforced when looking at when the graveyard curve stops tracking the best known curve in Figures 4.18-4.20.

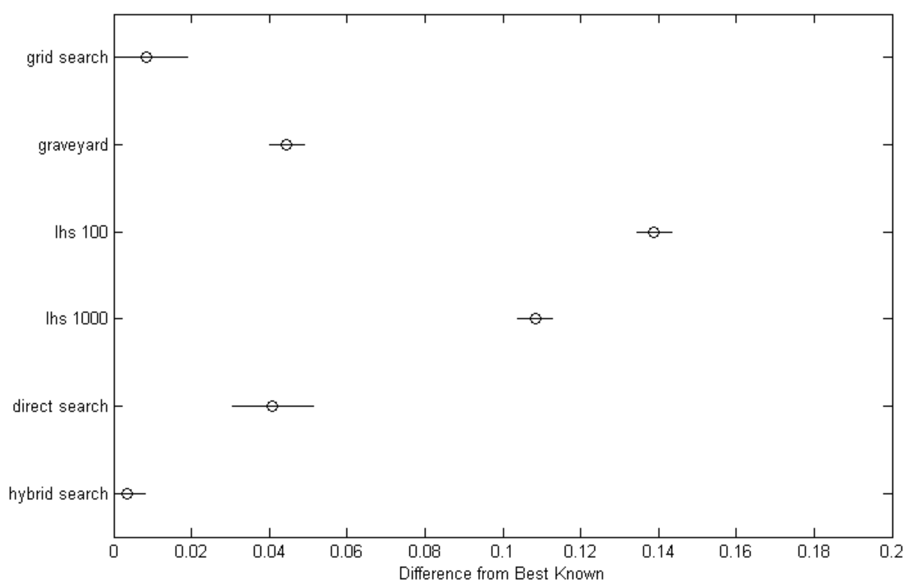


Figure 4.21 Error for Point of Interest #1

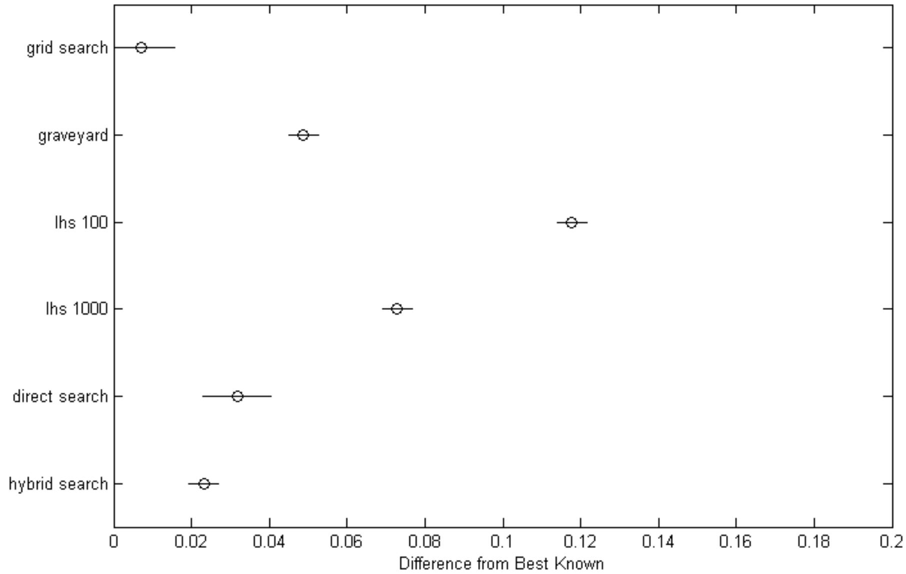


Figure 4.22 Error for Point of Interest #1

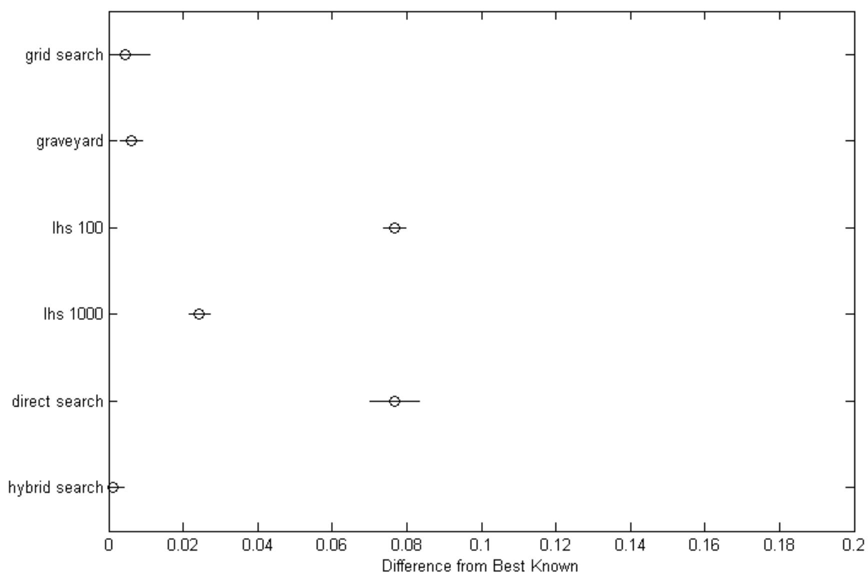


Figure 4.23 Error for Point of Interest #1

4.3.2.3 Computational Cost vs. Error

To visualize the trade-offs in computational cost and error, bubble plots of the six search techniques are shown in Figures 4.24-4.26. These bubble plots show that three of the six search techniques are located on the front of the trade-off curve in at least one of the figures. Again, the two search techniques that use the graveyard (graveyard and hybrid) are part of this group and are the only search techniques not dominated by another technique in at least one of the three figures.

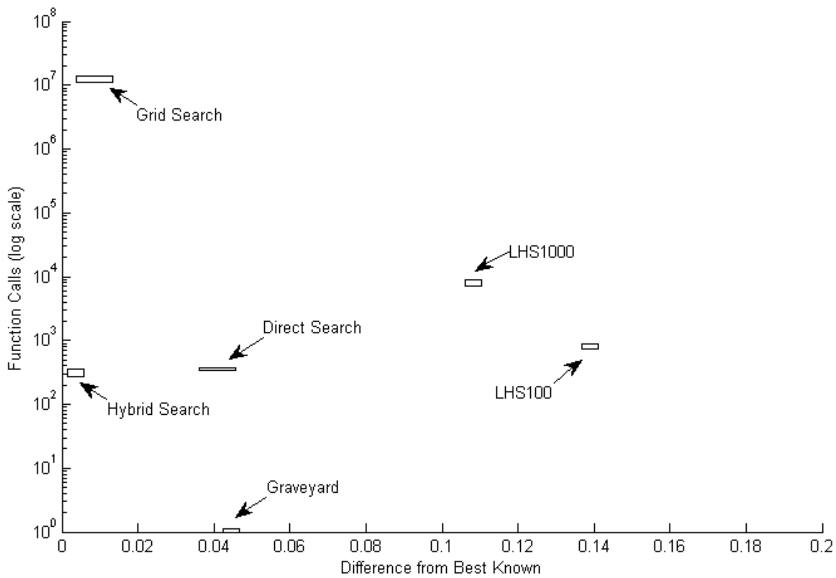


Figure 4.24 Tradeoffs for Point of Interest #1

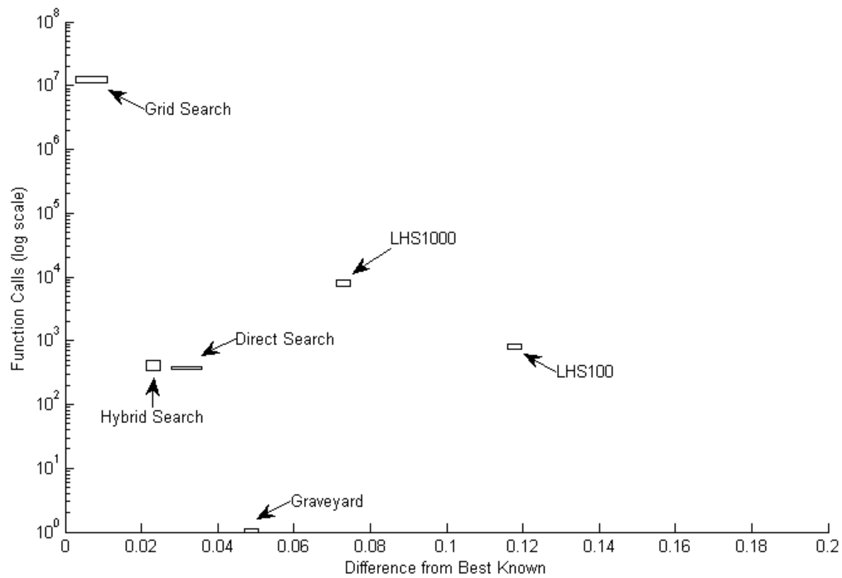


Figure 4.25 Tradeoffs for Point of Interest #1

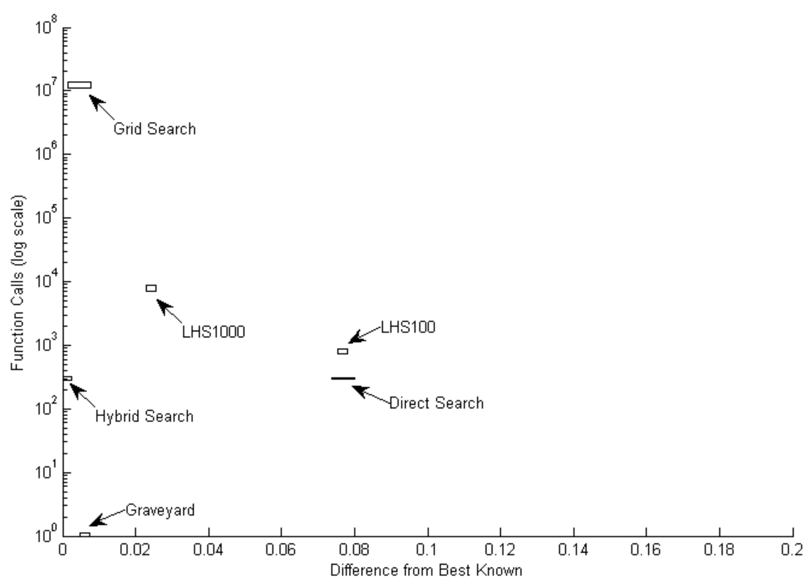


Figure 4.26 Tradeoffs for Point of Interest #1

4.4 Results

In this chapter, both the error and the computational cost of five search techniques were explored. More specifically, this chapter investigated which search techniques could be used to perform MGA in a more computationally efficient manner. Initial results from two case study problems indicate that using data from the graveyard leads to increased performance. In both case studies the graveyard and hybrid search, which makes use of the graveyard, outperform other techniques in both computational cost and error. The error from using the graveyard ranged from 0.01 to 0.06 with a mean, median, and mode all around 0.05, while the error from the hybrid search ranged from 0 to 0.02 with a mean, median, and mode all around 0.005. These numbers are fairly small, especially once you consider the minimum distance of both problems. The minimum distance in the three variable two bar truss case is $\frac{\sqrt{3}}{2}$ or 0.87. Additionally, the minimum distance in the four variable I-beam case is $\frac{\sqrt{4}}{2}$ or 1. This puts the maximum estimated error of the graveyard at 7% for the two bar truss and 6% for the I-beam. The hybrid comes in lower at around 2% maximum estimated error for both cases. The cost of performing a hybrid search on the two objective problems was around 500 functional evaluations, while the cost of using the graveyard is zero functional calls. Both of these were substantially less than the grid search, which was the original search technique used in chapter 3.

The grid search demonstrated that it is the most consistently accurate method available. It was the only search technique with error consistently near 0%. However, this reliability comes with extremely high computational expense. Both Latin hypercube samplings performed poorly in computational cost and error. Yet, a tradeoff between computational cost and error was observed. This means a designer using a latin hypercube sample could expect improved accuracy increase by simply increasing the sample size.

The final search technique to discuss is the direct search. This approach performed similarly to the hybrid search as it is based on the same optimization algorithm. However, it struggled to overcome locally optimal alternative designs. The hybrid search overcomes this shortfall by changing its starting point to a design that is theoretically closer to a global optimum.

In conclusion, the two search techniques that demonstrate the best tradeoffs between computational cost and error are the hybrid search and the graveyard. While the error within the graveyard can be reduced by limiting the amount of performance freedom, the error in the hybrid search is fairly stable for all performance freedom ranges.

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

5.1 Thesis Summary

In the previous four chapters, the expansion of an alternative generation technique to nonlinear multiobjective engineering design problems was explored. The first chapter served to introduce the motivation for expanding MGA and outline the research questions. The primary motivator for using MGA is to empower the designer to make better design decisions. Ideally, MGA would accomplish this by enhancing the design by shopping paradigm that is gaining popularity as a design decision tool. Recall that design by shopping allows a designer to form their preferences after visualizing the design space. MGA would enhance this visualization by allowing several similarly performing, but unique designs to be shown in addition to the Pareto frontier that is currently shown. Next, two research questions were posed. The first research question aimed to assess the effects of objective function formulation, while the goal of the second research question test whether graveyard data could be used to increase the computational efficiency of MGA searches in multiobjective problems.

The second chapter provided background into both related and supporting research as well as detailed further benefits of MGA. The related research fell largely into three different categories which included overcoming modeling uncertainties through the use of a join-cognitive system, decreasing the convergence time in concurrent design by identifying a range of solutions, and locating design freedom. The supporting research was focused mainly on different approaches that could be used as search techniques and also introduced a primer on one way analysis of variance. The benefits of MGA, versus similar methods, included improving the visualization of design by shopping, making use of existing datasets, reducing the computational cost of identifying design freedom, and being relatively easy to customize.

The third chapter explored the first research question. To do this, two case study problems were used. The first case study problem was a nonlinear single objective problem that dealt with the design a tripod. Meanwhile, the second case study problem was a nonlinear multiobjective problem that focused on the design of a vibrating platform. The reason for including a single objective problem into the research was twofold. First, it provides a case study that is easier for the reader to understand and visualize. Second, it decoupled the effects of the nonlinear aspects of the problem from the multiobjective aspects of the problem.

The fourth chapter explored the second research questions. Again, this used two case study problems. This time both problems were multiobjective. However, the problems were different in terms of the amount of design freedom they presented near the Pareto frontier as previously indicated by Simov [50-51]. The first problem dealt with the design of a two bar truss, while the second looked at the design of an I-beam.

This chapter serves to summarize the results of this thesis. In the next section the research questions identified in chapter 1 will be revisited. The results from chapters 3 and 4 will then be applied to answer the questions. Next, areas of future research that arise from this thesis will be discussed. The thesis will then conclude with final remarks.

5.2 Addressing the Research Questions

In this section, the research questions identified in chapter 1 are revisited. The goal is to evaluate the extent to which they have been answered in this thesis. Recall, that two research questions were posed and each will be examined separately below.

5.2.1 Research Question #1:

How does the definition of the objective function affect the alternative designs found by MGA in a multiobjective nonlinear engineering design problem?

To address research question #1, chapter 3 applied different distance metrics to two different case studies. Recall that distance metrics are only one type of objective function that can be used within MGA. Others include measures of variation and the assignment of random variable weights. Immediately, an application of MGA to the single objective tripod design problem illuminated an undesired behavior stemming from base MGA approach. This behavior involved how the previous MGA approach dealt with the combination of distance metrics to enable multiple alternative designs to be found. The prior approach suggested that summing the distances to each existing alternative design would produce useable results, however application of this approach yielded alternatives designs that clustered around the extremes (corners) of the design space. To encourage the alternatives to disperse more evenly throughout the design space, an approach that maximized the smallest distance to an existing alternative was adopted. This approach was applied to both test problems in chapter 3. The results suggest that maximizing the distance to the closest alternative provides a much more uniform representation of the design space than summation of the distances when searching for multiple alternatives in a nonlinear engineering design problem.

Beyond this, two distance metrics and three scaling techniques were tested. It was shown that the 1-norm and 2-norm distance metrics were not able to identify same alternative designs all the time. Further, it was noticed that alternatives found using a 1-norm distance metric seemed to be less influenced by the type of scaling technique chosen. When applied to a mix integer problem, both distance metrics failed to handle an integer choice variable properly. To fix this a hamming distance was needed to convert the variable into an integer value corresponding with the amount of change the variable represented.

Of the scaling techniques, max step scaling was the most consistent at expanding the feasible design space quickly. However, the performance of max step scaling was not substantially better than either bound scaling or percent difference scaling. Additionally, the bound scaling method gave some indication that it is able to increase the number of common designs found by the 1-norm and 2-norm distance metrics.

Lastly, testing with various levels of performance freedom on the two objective vibrating platform problem yielded interesting results. The results of this study gave an indication of correlation between the number of feasible designs and the number of common designs discovered. More precisely, as the number of feasible designs decrease, more common designs were found using different MGA settings. Further, an increased effect on the tails of the Pareto frontier due to changes in performance freedom was noticed.

To summarize, changes in the objective function formulation can cause different alternative designs to be discovered. In an effort to increase the repeatability and validity of future uses of MGA on multiobjective nonlinear engineering design problems a few suggestions are presented. First, it is suggested that designers us the minimum distance approach to multiple alternative searches instead of the summation approach. Second, 1-norm distance is suggested for problems where the user is unsure of the scaling. Further, 2-norm distance is suggested for problems where the user is able to use bound or max-step scaling and wants to use the distance measure in other mathematical equations. Hamming distance is recommended only for integer based choice variables. Finally, of the scaling techniques bound scaling is recommended if the user can supply good variable bounds. If variable bounds are not supplied or able to be solved for, then it is suggested that the designer use max step scaling. However, if the designer cannot afford the additional cost of max step scaling, Percent difference scaling is suggested. Choosing to not scale the variables is not recommended and should only be done when all the variables are very close in magnitude.

5.2.2 Research Question #2:

Can graveyard data be used to perform MGA in a computationally efficient manner on multiobjective nonlinear engineering design problems?

To answer this research question six search techniques were applied to two nonlinear multiobjective engineering design problems. The problems had different amounts of design freedom available near the Pareto frontier. This was done to illuminate any bias that the search techniques may have towards design freedom availability.

The six search techniques were comprised of four population based search techniques and two single point based search techniques. The population based techniques comprised of a graveyard, a grid search, and two latin hypercube samples of different size. The single point based search techniques were both based off of the Nelder-Mead simplex as it doesn't require derivative information to function. The two single point based search techniques varied according to which starting point they used. The direct search used the point of interest as a starting point, while the hybrid search used the most unique alternative from the graveyard as its starting point.

With each case study problem, three different tests were performed. The first test looked at how each search technique performed as design freedom was allowed to change. The results show that several of the search techniques experience an increase in error as the design freedom is increased. However, the uniqueness of the designs (as measured by the distance measure) also increases as the design freedom increases for all the search techniques.

The second test looked to see if the amount of error between each search technique was statistically equivalent. The results indicate that there is a 0% chance that the error of all six techniques is equivalent. Of the techniques, both single point based techniques consistently exhibit little error. Meanwhile, of the population based techniques, only the grid search exhibited consistently small error. In fact, the error exhibited by the grid search was routinely smaller than the error exhibited by either of the single point based search techniques.

The third and final test looked at computational cost in addition to the error of each method. With this new information the grid search was shown to have computational costs several orders of magnitude larger than the closest competitor. Of the remaining search techniques both the graveyard and the hybrid exhibited the best balance of computational cost and error. The maximum estimated error of the hybrid approach was 2%, while the maximum estimated error of the graveyard was 7%. Meanwhile, the cost for both was around 500 functional calls for the hybrid search and 0 functional calls for the graveyard.

In conclusion, the results indicated that using graveyard data does allow for MGA to be performed in a computationally efficient manner. In fact this thesis would recommend using either the graveyard or the hybrid search (both of which use graveyard data) over any of the other search techniques test. By providing more than one search technique that uses graveyard data, the designer is able to choose the technique that best meets their needs. In either case the graveyard search has be performed first. Again, the graveyard search provides results that are fairly accurate without needing any additional functional calls. However, if the user desires higher accuracy and can afford the additional functional calls to get them, they can make use of the hybrid search to achieve it.

5.3 Future Work

This thesis lays the groundwork for making MGA a viable nonlinear multiobjective engineering design tool. Throughout this paper opportunities for future avenues of research have been discovered. These opportunities are discussed in further detail in this section.

5.3.1 Surrogate Modeling Search Technique

As noted earlier, population based techniques tend to produce more error than their single point based counterparts. However, population based searches are far more robust than single point based search. The reduction in accuracy for population based searches comes from their lack of density near the point of interest. To achieve higher density requires larger sample sizes. The graveyard gets around this by being dense only near the optimal region of the performance space; however, it still produces more error on average than either of the single point based searches.

A promising compromise to increase the accuracy of population based searches without dramatically increasing the sample size comes from meta-modeling [64]. Meta-modeling, or surrogate modeling as it's sometimes called, is a type of curve fitting approach used to reduce the computational cost of complex optimization problem. Applying a meta-model to the graveyard data could conceivably allow for very accurate results with little to no additional computational cost. Once a model was fit, the hybrid search could be performed using the meta-model instead of the actual model of the system. While this may result in a similar number of functional calls, the cost would be dramatically reduced as the cost to calculate the value of a point on a curve is much smaller than the cost to calculate the performance of a design in a complex system.

5.3.2 Develop Tradeoff Curves for Population Based Search Techniques

Within Chapter 4 a tradeoff within the two differently sized latin hypercube samples was noticed. The latin hypercube sample with the larger sample size was always more accurate than the latin hypercube sample with the smaller sample size. Additionally, it was noted that the grid search produced very accurate results, but at extremely high computational cost. It is expected that the grid search would follow a similar tradeoff scheme as the latin hypercube sampling. More simply, it is expected that as the sample size of a population based search decreases the error will subsequently increase.

Currently, there are two recommended search techniques for MGA (graveyard and hybrid search). It is conceivable that at the proper settings the graveyard could be computationally viable and accurate. This means that there is an opportunity to measure the tradeoff that occurs between sample size and error in a grid search.

Additionally, a similar tradeoff curve could be made for the latin hypercube sampling technique as well as any future population based search techniques. This would allow future users of MGA to have a quick reference as to what sample size they should use for a given problem if they wanted to use a certain population based search technique instead of the recommended techniques.

5.3.3 Further Investigation of 1-Norm Distance

It was previously stated that alternatives found using a 1-norm distance metric seemed to be less affected by scaling than those found using a 2-norm distance metric. This is desirable trait if true; however, this thesis did not try to prove or disprove this discovery as it was outside its scope. Future research should be able to focus on either proving or disproving the hypothesis that alternatives found using a 1-norm distance metrics are less affected by scaling than those found using a 2-norm distance metric.

5.3.4 Optimal Number of Alternative Designs

Throughout this thesis, a set number of alternative designs were found. However, the chosen values have no special properties. It is suggested that research into the optimal number of alternative designs be performed.

It is hypothesized that the number of optimal alternative designs will depend on the number of design variables. By optimizing the number of alternative designs, the designer can ensure that they are capturing a large amount of design data without expending excessive computational resources. For instance, in Figure 3.15 the optimal number of designs would likely be three, as that is where the amount of additional design growth seems to begin to level off.

5.3.5 Reconfigurable System Design

A final opportunity comes in the form of using MGA to identify regions suitable for reconfigurable system design. Reconfigurable designs have been shown to offer an increase in performance over static designs [65]. Instead of forcing the designer to trade off performance between objectives, they can theoretically have a system that can achieve high levels of performance in multiple objectives at different times. Reconfigurable systems accomplish this by changing or reconfiguring their design variable values during operation. In an effort to reduce complexity of these systems it is desirable to limit the amount of variable change needed to achieve different performance states. It is thought that MGA could be used to identify these regions. Instead of using MGA to identify the design freedom associated with a point in the performance space, which MGA does throughout this thesis, it would identify points in the design space associated with large variations in the performance space.

5.4 Concluding Remarks

The work presented in this thesis is aimed at empowering the designer's ability to make better design decisions. By exploring the design space prior to making a decision the designer is claiming more control over the design process. Currently, too much of that control is put in the hands of computer algorithms. Despite the concerns raised in the previous work [6], this thesis demonstrates that MGA process can be used to identify alternative designs in a nonlinear multiobjective engineering design problem. The research verifies that the formulation of the objective function used in MGA will affect the alternatives identified. Moreover, this research demonstrates that this process can be done in a computationally efficient and accurate manner through the help of graveyard data.

REFERENCES

[1] D. D. Woods, “Paradigms for intelligent decision support” in Intelligent Decision Support in Process Encvironments. E. Hollnagel. G. Mancini, and D. D. Woods, Eds. New York: Springer-Verlag.1986.

[2] Brill, E.D., “The use of optimization models in public-sector planning”, Management Science, vol. 25, 1979, pp. 413–422.

[3] Brill, E.D., Chang, S., and Hopkins, L.D., “Modeling to Generate Alternatives: The HSJ Approach and an Illustration Using a Problem in Land Use Planning”, Management Science, vol. 28, Mar. 1982, pp. 221-235

[4] S. Chang, E. D. Brill, Jr. and L. D. Hopkins, “Efficient random generation of feasible alternatives: A land use example,” J. Regional sci,, vol 22, no. 3, pp. 303-313, Aug. 1982

[5] Chang, S.Y., Brill, E.D., and Hopkins, Lewis D., “Modeling to generate alternatives: A fuzzy approach,” Fuzzy Sets and Systems, vol. 9, 1983, pp. 137–151.

[6] Kripakaran, P., and Gupta, A., “MGA–A Mathematical Approach to Generate Design Alternatives,” Lecture Notes in Computer Science, vol. 4200, 2006, p. 408.

[7] Simulation-Optimization Framework to Support Sustainable Watershed Development by Mimicking the Predevelopment Flow Regime Laurel Reichold, Emily M. Zechman, E. Downey Brill, and Hillary Holmes, J. Water Resource Plann. Manage. 136, 366 (2010)

[8] Joseph F. DeCarolis, Using modeling to generate alternatives (MGA) to expand our thinking on energy futures, Energy Economics, Volume 33, Issue 2, March 2011, Pages 145-152, ISSN 0140-9883

[9] Deb, K., Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, 2009.

[10] Zadeh, L. 1963: Optimality and Non-Scalar-Valued Performance Criteria. IEEE Trans Autom Control 8, 59–60

[11] Sullivan, Lawrence P. (1986), "Quality Function Deployment," Quality Progress, 19, 6 (June), 39-50

[12] See, T.K., Gurnani A., and Lewis, K., 2005, “MultiAttribute Decision Making Using Hypothetical Equivalents and Inequivalents,” ASME Journal of Mechanical Design, 126(6), pp. 950-958

[13] See, T.K., and Lewis, K., 2005, “A Decision Support Formulation for Design Teams: A Study in Preference Aggregation and Handling Unequal Group Members,” ASME Design Technical Conferences, Design Automation Conference, DETC2005-84766.

[14] Loughlin, Daniel H.; Ranjithan, S. Ranji; Brill, E. Downey; Baugh, John W.. "Genetic Algorithm Approaches For Addressing Unmodeled Objectives In Optimization Problems" Engineering Optimization 33.5 (2001). 11 Apr. 2011

[15] Ferguson, S., Malegaonkar, P., Olewnik, A., Cormier, P., and Kansara, S., 2010, “Mass Customization: A Review of the Paradigm Across Marketing, Engineering and Distribution Domains,” ASME Design Engineering Technical Conference, Design Automation Conference, Montreal, Quebec, DETC2010-28753.

[16] Pareto, V., Schwier, A.S., and Page, A.N., Manual of political economy, Kelley (Augustus M.) Publishers, US, 1971.

[17] Vanderplaats, G., Multidiscipline design optimization, Monterey Calif.: Vanderplaats Research & Development Inc., 2007.

[18] Balling, R., 1999, “Design by Shopping: A New Paradigm,” Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization, 295-297.

[19] Pine, B.J., and Davis, S., Mass Customization: The New Frontier in Business Competition, Cambridge , MA : Harvard Business School Press, 1993

[20] Franke, N., Schreier, M., “Product uniqueness as a driver of customer utility in mass customization”, Marketing Letters, 2008

[21] Development of a Functional Basis for Design Robert B. Stone and Kristin L. Wood, J. Mech. Des. 122, 359 (2000)

[22] Finch, W., and Ward, A. C., 1997, “A Set-based System for Eliminating Infeasible Designs in Engineering Problems Dominated by Uncertainty,” 1997 ASME Design Engineering Technical Conferences, Sacramento, CA., DETC97/DTM-3886.

[23] Chen, W. and Lewis, K., 1999, “A Robust Design Approach for Achieving Flexibility in Multidisciplinary Design”, AIAA Journal, 7(8), 982-989.

[24] Shan, S., and Wang, G., 2004, “Space Exploration and Global Optimization for Computationally Intensive Design Problems: A Rough Set Based Approach,” Structural and Multidisciplinary Optimization, 28: 427-441.

[25] Sobek, D., Ward, A. and Liker, J., 1999, Durward K. Sobek, Allen C. Ward and Jeffrey K. Liker "Toyota's Principles of Set-Based Concurrent Engineering" Sloan Management Review, Vol. 40, no. 2, pp. 67 – 83

[26] Wood, K. L., and Antonsson, E. K. Computations with Imprecise Parameters in Engineering Design: Background and Theory. ASME Journal of Mechanisms, Transmissions, and Automation in Design 111, 4 (Dec. 1989), 616-625

[27] Wood, K. L., and Antonsson, E. K. Modeling Imprecision and Uncertainty in Preliminary Engineering Design. Mechanism and Machine Theory 25, 3 (Feb. 1990), 305-324. Invited paper.

[28] Wood, K. L., Otto, K. N., and Antonsson, E. K. Engineering Design Calculations with Fuzzy Parameters, Fuzzy Sets and Systems 52, 1 (Nov. 1992), 1-20

[29] Fogel, L. J., A. J. Owens, and M. J. Walsh. (1966). Artificial Intelligence through Simulated Evolution. New York: Wiley.

[30] Rechenberg, I. (1965). Cybernetic Solution Path of an Experimental, Problem Library Translation No. 1122. Farnborough, UK: Royal Aircraft Establishment, Ministry of Aviation.

[31] Holland, J. H. (1975). Adaptations in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press.

[32] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley.

[33] Koza, J. R. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press.

[34] Xiao, Ningchuan "An Evolutionary Algorithm for Site Search Problems". Geographical Analysis (0016-7363), 38 (3), p. 227.

[35] Xiao, N., D. A. Bennett, and M. P. Armstrong. (2002). ‘‘Using Evolutionary Algorithms to Generate Alternatives for Multiobjective Site Search Problems.’’ Environment and Planning A 34(4), 639–56.

[36] Moller, B., Reuter U. Uncertainty forecasting in engineering. Berlin: Springer; 2007.

[37] Alefeld G, Herzberger J. Introduction to interval computations. New York: Academic Press; 1983.

[38] Moller, B., Beer, M., Engineering computation under uncertainty - Capabilities of non-traditional models, Computers & Structures, Volume 86, Issue 10, Uncertainty in Structural Analysis - Their Effect on Robustness, Sensitivity and Design, May 2008, Pages 1024-1041, ISSN 0045-7949

[39] Madhavan, K., Shahan, D., Seepersad, C.C., Hlavinka, D., Benson, W., 2008, "An Industrial Trial of a Set-Based Approach to Collaborative Design", ASME IDETC/CIE, Brooklyn, NY, Paper Number: DETC2008-49953.

[40] Chen, Wei, Sahai,A. Messac,A. and Sundararaj,G., 2000. "Exploration of the Effectiveness of Physical Programming in Robust Design." Journal of Mechanical Design 122: 155–163.

[41] Messac, A., Martinez, M., and Simpson, T., "Introduction of a Product Family Penalty Function Using Physical Programming." Journal of Mechanical Design 124 (2002): 164–172.

[42] Malak, R., 2008. "Using parameterized efficient sets to model alternatives for systems design decisions." - Georgia Tech's Institutional Repository. PhD thesis, Georgia Tech

[43] Shir, O., Preuss, M., Naujoks, B., and Emmerich, M., 2009. "Enhancing Decision Space Diversity in Evolutionary Multiobjective Algorithms." Evolutionary Multi-Criterion Optimization. Ed. Volume 5467 . Berlin, Heidelberg: Springer Berlin Heidelberg, chapter 12, 95–109.

[44] Toffolo, A. and Benini, E., 2003. "Genetic Diversity as an Objective in Multi-Objective Evolutionary Algorithms." Evolutionary Computation 11: 151–167.

[45] G. G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design optimization," Journal of Mechanical Design, vol. 129, no. 4, pp. 370–380, 2007.

[46] Azarm, S., Reynolds, B. J., and Narayanan, S., 1999, "Comparison of Two Multiobjective Optimization Techniques with and within Genetic Algorithms," 1999 ASME Design Engineering Technical Conferences, Las Vegas, NV, DETC99/DAC-8584.

[47] Cacuci, D. G., Ionescu-Bujor, M., and Navon, I, 2003. Sensitivity and uncertainty analysis. Boca Raton: Chapman & Hall/CRC Press.

[48] Saltelli, A. et al. 2008. Global Sensitivity Analysis. The Primer. John Wiley & Sons, Ltd.

[49] Lee S.M, Goal Programming for Decision Analysis (Auerbach, Princeton, NJ, 1972).

[50] Simov, P., and Ferguson, S., 2010, "Investigating the Significance of 'One-to-Many' Mappings in Multiobjective Optimization," ASME Design Engineering Technical Conference, Design Automation Conference, Montreal, Quebec, DETC2010-28689.

[51] Simov, P. Investigating the Significance of "One-to-many" Mappings in Multiobjective Optimization. Thesis. North Carolina State University, 3-Nov-2010. Print.

[52] "Chapter 11.6 - Nelder-Mead (Simplex) Method." *MultiForte and Boomer Manual*. Web. http://www.boomer.org/c/p3/c11/c1106.html.

[53] "Latin Hypercube Sample - MATLAB." *MathWorks - MATLAB and Simulink for Technical Computing*. Web. http://www.mathworks.com/help/toolbox/stats/lhsdesign.html

[54] I. D. Coope and C. J. Price. On the convergence of grid-based methods for unconstrained optimization. *SIAM Journal on Optimization*, 11(4):859–869, 2001.

[55] Navidi, William Cyrus. "Factorial Experiments." *Statistics for Engineers and Scientists*. Boston, MA: McGraw-Hill, 2006. Print.

[56] Arora, J., Introduction to Optimum Design, Academic Press, 2004.

[57] Deza, M.M., and Deza, E., Encyclopedia of Distances, Springer, 2009.

[58] Black, P.E., "Manhattan distance", in Dictionary of Algorithms and Data Structures [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. http://www.itl.nist.gov/div897/sqg/dads/HTML/manhattanDistance.html

[59] "Find Minima of Multiple Functions Using Genetic Algorithm" MathWorks - MATLAB and Simulink for Technical Computing. www.mathworks.com/help/toolbox/gads/gamultiobj.html

[60] Foster, G., and Ferguson, S., 2010, "Exploring the Impact of Distance Metrics on Alternative Generation in a Multiobjective Problem," 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX, AIAA-2010-9091.

[61] Beiser, Arthur. Essential Math for the Sciences: Algebra, Trigonometry, and Vectors. New York: McGraw-Hill, 1969. Print.

[62] "Find Minimum of Unconstrained Multivariable Function Using Derivative-free Method" MathWorks - MATLAB and Simulink for Technical Computing. 25-Feb-2011 www.mathworks.com/help/techdoc/ref/fminsearch.html

[63] Hacker, K., and Lewis, K., 2002, "Robust Design through the Use of a Hybrid Genetic Algorithm," 2002 ASME Design Engineering Technical Conferences and Computers and Information in Engineering, Montreal, Canada, DETC2002/DAC-34108

[64] Viana, F.A.C. Gogu, C. and Haftka, R.T. "Making the most out of surrogate models: tricks of the trade," ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Montreal, Canada, August 16 - 18, 2010, DETC2010-28813.

[65] Sullivan, E., Tortorice, M., and Ferguson, S., 2010, "Using Design Reconfigurability to Mitigate the Effects of Uncontrolled System Variations," 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX, AIAA-2010-9185.