The Pennsylvania State University

The Graduate School

College of Engineering

ASSESSMENT OF USER-GUIDED VISUAL STEERING COMMANDS DURING TRADE SPACE EXPLORATION

A Thesis in

Mechanical Engineering

by

Daniel E. Carlsen

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2008

The thesis of Daniel E. Carlsen was reviewed and approved* by the following:

Timothy W. Simpson
Professor of Mechanical and Industrial Engineering
Thesis Adviser

Mary Frecker
Professor of Mechanical Engineering

Karen A. Thole
Professor of Mechanical Engineering
Head of the Department of Mechanical and Nuclear Engineering

*Signatures are on file in the Graduate School.

ABSTRACT

Trade space exploration is a promising decision-making paradigm that provides a visual and more intuitive means for formulating, adjusting, and ultimately solving design optimization problems. This is achieved by combining multi-dimensional data visualization techniques with visual steering commands to allow designers to “steer” the optimization process while searching for the best, or Pareto optimal, design(s). In this thesis, results from an empirical assessment of the performance of these visual steering commands are presented. This is done by performing a study that compares the performance of different combinations of these visual steering commands to automated samplers, including a multi-objective genetic algorithm, that are executed “blindly” on the same design problems with no human intervention. The resultant Pareto frontiers generated by the combinations of visual steering commands and automated samplers are compared to one another using the ε-performance metric to assess the extent to which they have identified the reference (or best known) Pareto frontier. Specifically, three test problems are examined: (1) a sandwich beam, (2) an aircraft wing, and (3) a vehicle configuration problem. The results of this study indicate that the visual steering commands – depending on the complexity of the test problem – can provide a 3x - 6x increase in the number of Pareto solutions that are obtained when the human is “in-the-loop” during the optimization process compared to an automated sampler. The improvements are even more dramatic in cases where automated samplers have a difficult time finding feasible solutions. In addition user-guided trials can provide from a 10x - 32x increase in the number of Pareto solutions obtained over random searching. As such,

this study provides empirical evidence of the benefits of interactive visualization-based strategies to support engineering design optimization and decision-making.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

Part of this work, namely, parts of the introduction, review of related work, overview of the visualization software, and some of the experimental setup and results of the vehicle configuration model experiment, comes from work done for a paper appearing in the 2008 ASME International Design Engineering Technical Conferences (IDETC). This thesis is an extension of that work in that it provides further support for the vehicle configuration model experiment results and demonstrates the usefulness of visual steering commands in building a suitable Pareto frontier on additional problems of varying complexity.

As first author of the aforementioned paper, the author of this thesis contributed in a number of ways. The first was helping to write the background information used in the paper. The main contributions came from creating and running the user-guided trials, assisting the other user with creating a second set of data for those trials, completing the analysis done on the user-guided trials, and establishing the conclusions and suggestions for future work. In addition, the author reviewed and revised the paper as a whole.

ACKNOWLEDGEMENTS

I would like to thank my adviser, Dr. Timothy W. Simpson, for the direction and help he provided in all aspects of this work. I am also grateful to Dr. Mary Frecker for serving as my thesis reader and for reviewing my thesis on such short notice. I would also like to thank Matthew Malone and Joshua Kollat, students at The Pennsylvania State University, for their assistance in running user-guided trials and establishing the use of the Pareto frontier performance metric. In addition, I would like to thank Scott Ferguson for his assistance with the vehicle configuration model and the results from the exhaustive MOGA. Finally, I would like to thank my fiancé for her constant support and my parents for their encouragement in pursuing my academic goals.

This work has been supported by the National Science Foundation under NSF Grant No. CMMI-0620948. Any opinions, findings, and conclusions or recommendations presented in this thesis are those of the author and do not necessarily reflect the views of the National Science Foundation.

CHAPTER 1

INTRODUCTION

Many engineering designers employ optimization-based tools and approaches to help them make decisions particularly during the design of complex systems such as automobiles, aircraft, and spacecraft, which require tradeoffs between multiple conflicting and competing objectives. Trade space exploration is a promising alternative decision-making paradigm that provides a visual and more intuitive means for formulating, adjusting, and ultimately solving design optimization problems. Trade space exploration is an embodiment of the Design by Shopping paradigm advocated by Balling [1]: designers want to "shop" to gain intuition about trades, what is feasible and what is not, and to learn about their alternatives first before making their decisions. Balling noted that the traditional optimization-based design process of "1) formulate the design problem, 2) obtain/develop analysis models, and 3) execute an optimization algorithm" often leaves designers unsatisfied with their results because the problem is usually improperly formulated: "the objectives and constraints used in optimization were not what the owners and stakeholders really wanted…in many cases, people don't know what they really want until they see some designs" [1]. Studies in other fields have resulted in similar findings. For example, Wilson and Schooler [2] noted that people do worse at some decision tasks when they are asked to analyze the reasons behind their preferences or assess all of the attributes of their choices. Likewise, Shanteau [3] observed that when people are dissatisfied with the results of a rational decision making process, they often change their ratings to achieve their desired result [4].

In order to support this trade space exploration, researchers at the Applied Research Laboratory (ARL) and Penn State have developed the ARL Trade Space Visualizer (ATSV) [4,5,6], a Java-based application that is capable of visualizing multi-dimensional trade spaces using glyph, 1-D and 2-D histogram, 2-D scatter, scatter matrix, and parallel coordinate plots, linked views [7], and brushing [8]. Figure 1 shows several examples of its multi-dimensional data visualization capability. The glyph plot (left) can display up to seven dimensions by assigning variables to the x-axis, y-axis, z-axis, text labels, size, color, orientation, and transparency of the glyph icons. The scatter matrix (top right), a grid of all 2-D scatter plots, is useful for visualizing trends and two-way interactions in the data. Histograms (bottom right) show the distribution of samples in each dimension.

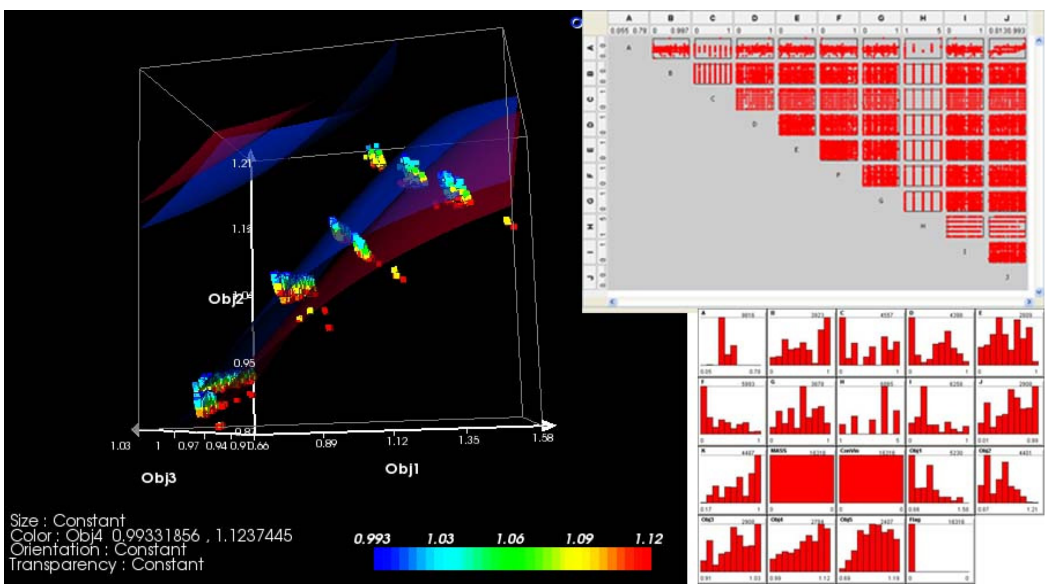


Figure 1. Three Displays of Data in ATSV

The design variable (input) and performance (output) data for different design alternatives can either be generated off-line and then input into ATSV for visualization and manipulation or it can be generated dynamically "on-the-fly" by linking a simulation model directly with ATSV using its Exploration Engine capability [4]. If the simulation model is too computationally expensive to be executed in real-time, then low-fidelity metamodels can be constructed and used as approximations for quickly searching the trade space [9].

Currently, there is a considerable amount of research in formalizing methods, tools, and procedures to support trade space exploration. Of particular interest for this thesis are interactive optimization-based methods, falling primarily into the area of *computational steering,* which gives the user (e.g., a designer) the ability to interact with a simulation during the optimization process to help "steer" or guide the search process toward what looks like an optimal solution. The designer is presented with a visual representation of the optimization process and then by using his/her intuition, heuristics, and/or some other methods, the designer adjusts the design space in order to move toward something that may not have been intuitive at the beginning of the simulation. For example, Wright, et al. [10] applied computational steering in designing the geometry and selecting the grade of glass for a furnace. Kesavadas and Sudhir [11] created large-scale manufacturing simulations by allowing users to make quick changes "on-the-fly" and then continue with the simulation. Messac and Chen [12] proposed an interactive visualization method wherein the progress of the optimization is visualized – but not steered – throughout the process. Finally, Visual Design Steering [13,14] allows users to stop and redirect the optimization process to improve the solution; however, their visualization capabilities are currently limited to 2-D and 3-D representations of constraints and objectives [4].

In line with this ongoing research to support trade space exploration, this thesis presents results from an empirical assessment of the performance of visual steering commands – visually-specified controls that allow designers to "steer" an optimization algorithm – that were developed previously [4]. This is done by performing a study that compares the performance of different combinations of these visual steering commands to automated samplers, including a multi-objective genetic algorithm, that are executed "blindly" on the same design problems with no human intervention. The resultant Pareto frontiers generated by the combinations of visual steering commands and automated samplers are compared to one another using the ε-performance metric to assess the extent to which they have identified the reference (or best known) Pareto frontier. Specifically, three test problems are examined: (1) a sandwich beam, (2) an aircraft wing, and (3) a vehicle configuration problem. Related research in computational steering is discussed next in Chapter 2 along with a review of the visual steering commands available in ATSV. Chapter 3 describes the test problems used in this work and the experimental set-up for this study. The results and findings are discussed in Chapter 4, and conclusions and future work are outlined in Chapter 5.

CHAPTER 2

REVIEW OF RELATED WORK AND OVERVIEW OF VISUAL STEERING COMMANDS

2.1 Related Work

Scott, et al. [15] recently proposed that including humans "in the loop" throughout the decision-making process improves the outcome of the process. They investigated the effects of integrating humans into the optimization process, and found that "combining the human's superior intelligence with the computer's superior computational speed can result in better solutions than either could produce alone". Additional advantages include learning about the problem and the interrelationships between objectives and having the ability to guide the solution process in a desired direction and possibly even changing his/her mind while learning [16]. Solutions generated through human interaction are better understood by the user than solutions merely given to them by an optimization algorithm. Moreover, the computational costs can be significantly reduced since only solutions of interest to the decision-maker are generated [15].

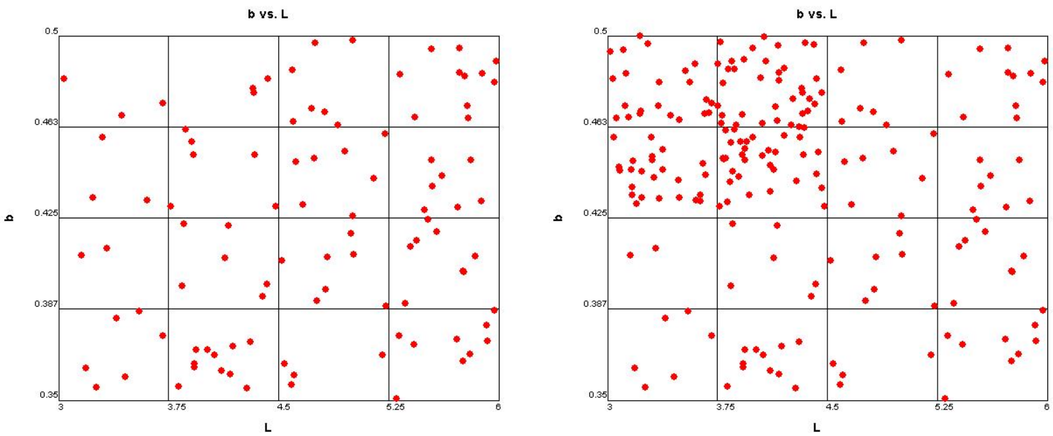
Madar, et al. [17] are investigating the effects of human interaction on a particular optimization algorithm, namely, particle swarm optimization. By using their visual, cognitive, and strategic abilities, human users can improve the performance of the computer search algorithm. Thus, interactive optimization approaches seek to combine expert knowledge with computational power. Michalek and Papalambros [18] propose in their work on architectural layouts that "the designer's interaction causes the program to dynamically change the optimization representation on-the-fly by adding, deleting, and modifying objectives, constraints, and structural units". Their "on-the-fly" methodology is applicable for architectural design, but the usefulness of it in complex system design conceptualization requires further exploration.

2.2 Overview of Visual Steering Commands

Visual steering commands help a designer to form their preferences while they explore (i.e., "shop") the trade space. They range in scope from commands used to sample the entire trade space to commands which focus around a region or point of interest. ATSV provides a suite of controls, including these visual steering commands, once an Exploration Engine is loaded, to help designers navigate and explore the trade space by (1) randomly sampling the design space, (2) searching near a point of interest, (3) searching in a direction of preference, or (4) searching for the Pareto frontier [4]. A brief summary [4] of each of these visual steering commands follows.

2.2.1 Design Space Sampler

Design space samplers are used to populate the trade space and are typically invoked if there is no initial data available or when the designer simply wants to explore the design space. The user can sample the design space manually using slider bar controls for each input dimension or randomly. When sampling randomly, the user specifies the number of samples to be generated and the bounds of the multi-dimensional hypercube of X. Monte Carlo sampling then randomly samples the inputs – drawing from a uniform, normal, or triangular distribution – and executes the simulation model, storing the corresponding output in the database. The bounds of the design variables can be reduced at any point to bias the samples in a given region if desired. An example is shown in Figure 2.

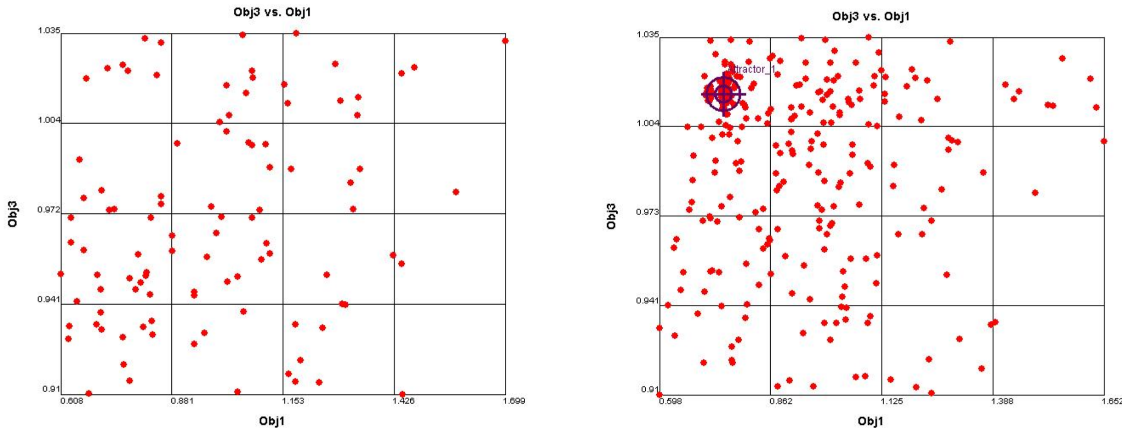


(a) 100 Initial Samples (b) 100 New Samples in a Reduced Region of Interest

Figure 2. Example of Design Space Sampler

2.2.2 Point Sampler

Point samplers, also referred to as attractors, are used to generate new sample points near a user-specified location in the trade space. The attractor is specified in the ATSV interface with a graphical icon that identifies an *n*-dimensional point in the trade space, and then new sample points are generated near the attractor – or as close as they can get to it. Unbeknownst to the user, the attractor generates new points using the Differential Evolution (DE) algorithm [19], which assess the fitness of each new sample based on the normalized Euclidean distance to the attractor. As the population evolves in DE, the samples get closer and closer to the attractor. An example is shown in Figure 3 where the user specifies an attractor to fill in a "gap" in the trade space (see Figure 3a). The new samples cluster tightly around Attractor_1 as seen in Figure 3b.

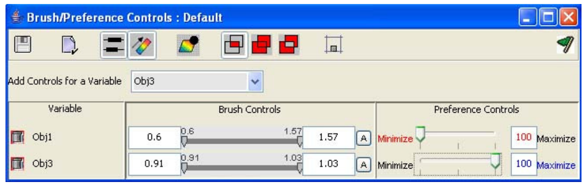


(a) 100 Initial Samples (b) New Samples Generated Near the Attractor

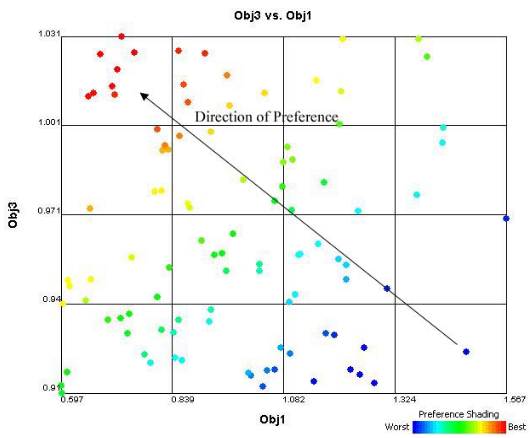
Figure 3. Example of Point Sampler using an Attractor

2.2.3 Preference-based Sampler

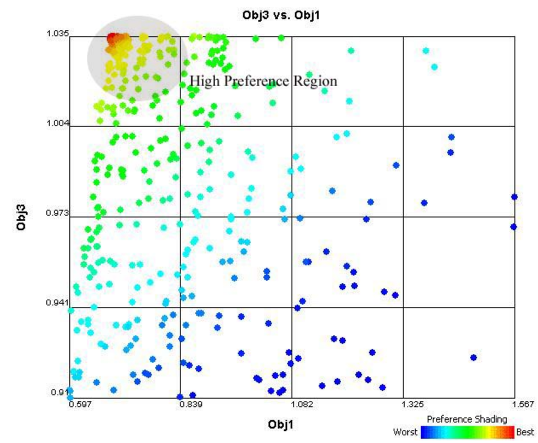
Preference-based samplers allow users to populate the trade space in regions that perform well with respect to a user-defined preference function. New sample points are also generated by the DE algorithm, but the fitness of each sample is defined by the user's preference structure instead of the Euclidean distance. An example of the preference-based sampler is shown in Figure 4. Using ATSV's brushing and preference controls, the user specifies a desire to minimize Obj1 and maximize Obj3 with equal weighting (see Figure 4a). Figure 4b shows the initial samples shaded based on this preference, and Figure 4c shows the new samples, where the concentration of points increases in the direction of preference, namely, the upper left hand corner of the plot.



(a) Brush Settings Indicating User Preference Structure



(b) Initial Samples Shaded Based on Preference

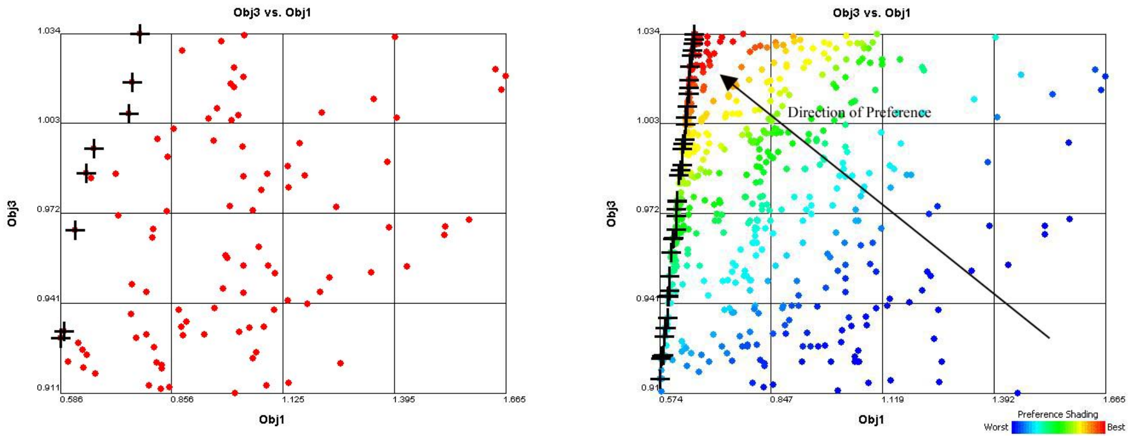


(c) New Samples Generated in Direction of Preference

Figure 4. Example of Preference-based Sampler

2.2.4 Pareto Sampler

Pareto samplers are used to bias the sampling of new designs in search of the Pareto frontier once the user has defined his/her preferences on the objectives. The DE algorithm is again used to accomplish this sampling but is modified to solve multi-objective problems [20]. An example of this sampler is shown in Figure 5. Using the same preference as before (i.e., minimize Obj1 and maximize Obj3 with equal weighting), Figure 5a shows the Pareto points in the initial samples while Figure 5b shows the Pareto frontier after executing 7 generations of the DE with a population size of 25 points. The points are also shaded to indicate the region of high (red) and low (blue) preference along the Pareto frontier.



(a) Initial Samples (Pareto Points Denoted by +) (b) New Samples Generated Along Pareto Frontier

Figure 5. Example of Pareto Sampler

These visual steering commands can be used together in any combination to explore the trade space. When used in concert with ATSV, designers have a powerful multi-dimensional visualization tool with the capability to "steer" the optimization process while navigating the trade space to find the best design. To determine the extent to which these visual steering commands are effective in locating good design solutions, the study described next in Chapter 3 is developed to compare the performance of different combinations of these visual steering commands to automated samplers, including a multi-objective genetic algorithm, that are executed "blindly" on the same design problems with no human intervention. The resultant Pareto frontiers generated by the combinations of visual steering commands and automated samplers are compared to one another by means of the ε-performance metric to assess the extent to which they have identified the reference (or best known) Pareto frontier. Specifically, three test problems are examined: (1) a sandwich beam design problem, (2) an aircraft wing design problem, and (3) a vehicle configuration problem.

CHAPTER 3

EXPERIMENTAL SET-UP AND TEST PROBLEMS

3.1 Test Problems

As stated previously, the objective in this thesis is to assess the performance of the visual steering commands discussed in Chapter 2 during trade space exploration. To do so, three test problems are examined, ranging in complexity from a two objective problem to a five objective problem with varying numbers of inputs and constraints, as summarized in Table 1. While stating general preferences in each of these problems beforehand may seem counter-intuitive to trade space exploration, the end goal in this study is to assess the effectiveness of the visual steering commands used in conjunction with ATSV in obtaining the Pareto frontier with a limited number of function evaluations and to demonstrate that doing so is more effective than simply allowing an automated search to run "blindly". Specifically, the problems examined consist of a sandwich beam [21] design problem, a wing [22] design problem, and vehicle configuration model [23,24,25]. Each problem is described in more detail next.

Table 1. Summary of Test Problems

Test Problem	Problem Formulation			Origin of Problem
	# Inputs	# Objectives	# Constraints	
Sandwich Beam	5	2	3	[21]
Aircraft Wing	6	3	3	[22]
Vehicle Configuration Model	11	5	1	[23,24,25]

3.1.1 Sandwich Beam Design Problem

The first test problem examined is a sandwich beam design problem developed by Messac [21]. This problem has five input design variables, two objectives, and three constraints. It consists of a motor vibrating on top of a supporting beam of length L, width b, and is symmetric about its center plane. The beam is made up of three layers of material with heights d_1, d_2, and d_3, as shown in Figure 6 and properties as shown in Table 2. The objective in this problem is to minimize the Cost and maximize the Fundamental Frequency subject to constraints on the mass of the beam and thickness of the material layers as well as bounds on the input variables.

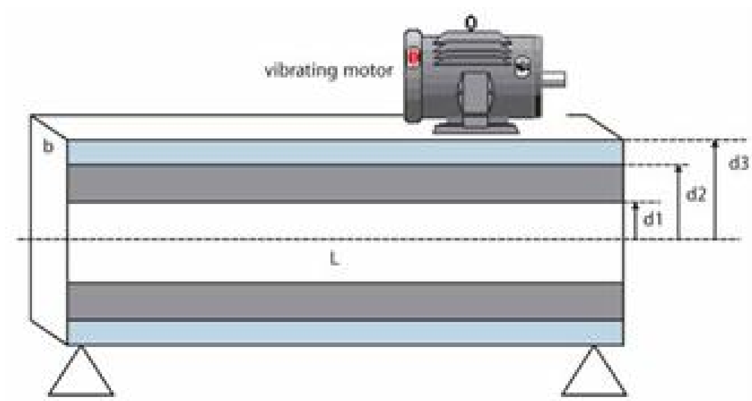


Figure 6. Sandwich Beam Design Variable Definition [26]

The sandwich beam problem can be summarized as follows:

Minimize: Cost (1)
Maximize: Fundamental Frequency

Subject to:
$$d_2 - d_1 \geq 0.01$$
$$d_3 - d_2 \geq 0.01$$
$$\text{Mass} \leq 2800$$

where:
$$3 \leq L \leq 6$$
$$0.05 \leq d_1 \leq 0.50$$
$$0.10 \leq d_3 \leq 0.55$$
$$0.20 \leq d_3 \leq 0.60$$
$$0.35 \leq b \leq 0.50$$

$$\text{Cost (\$)} = 2bL[c_1d_1 + c_2(d_2 - d_1) + c_3(d_3 - d_2)]$$

$$\text{Fundamental Frequency (Hz)} = (\pi/2L^2)(EI/\mu)^{1/2}$$
$$EI = (2b/3)\,[E_1d_1^3 + E_2(d_2^3 - d_1^3) + E_3(d_3^3 - d_2^3)]$$
$$\mu = 2b\,[\rho_1d_1 + \rho_2(d_2 - d_1) + \rho_3(d_3 - d_2)]$$

Table 2. Material Properties for the Sandwich Beam Design Problem

Material	Density, ρ (kg/m^3)	E (N/m^2)	c (\$/m^3)
Layer 1	100	1.60E+09	500
Layer 2	2770	7.00E+10	1500
Layer 3	7780	2.00E+11	800

3.1.2 Aircraft Wing Design Problem

The second problem used in this study is an aircraft wing design problem. The original problem created by Simpson and Meckesheimer [22] is a single objective problem with the goal to minimize Cost; however, for the purpose of this study it was modified to include two more objectives. The original problem has six bounded input design variables to size the wing, constraints on the Range, Buffet Altitude, and Takeoff Field Length, and as mentioned, the objective is to minimize Cost. The problem used in this case is modified by adding two objectives: (1) maximize Range and (2) minimize Takeoff Field Length. Figure 7 defines the input variables for sizing the wing. In the model, the output variables (Cost, Range, Buffet Altitude, and Takeoff Field Length) have been normalized to range from 0 to 1.

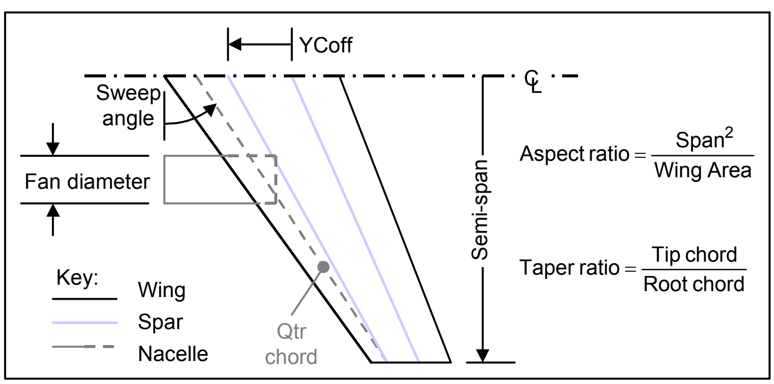


Figure 7. Wing Problem Input Variable Definition [22]

The wing design problem can be summarized as follows:

Minimize:	Cost	(2)
Maximize:	Range	
Minimize:	Takeoff Field Length	
Subject to:	Range ≥ 0.589	
	Buffet Altitude ≥ 0.603	
	Takeoff Field Length (TOFL) ≤ 0.377	
where:	$900 \leq$ Semi-span, $x_1 \leq 1150$	
	$8 \leq$ Aspect ratio, $x_2 \leq 13$	
	$31 \leq$ Sweep angle, $x_3 \leq 37$	
	$0.15 \leq$ Taper ratio, $x_4 \leq 0.25$	
	$0.75 \leq$ Y Coff, $x_5 \leq 1$	
	$80 \leq$ Fan Diameter, $x_6 \leq 90$	

Cost $= 0.2854 - 0.005x_6 + 0.3109x_5 - 0.0122x_3 - 0.2095x_4 - 0.4836x_2 + 0.4431x_1 + 0.1037x_6x_6 - 0.0592x_5x_5 - 0.0204x_3x_3 + 0.1057x_4x_4 + 0.2494x_2x_2 + 0.0218x_1x_1 + 0.0581x_6x_5 + 0.0025x_6x_3 + 0.0034x_6x_4 + 0.0502x_6x_2 - 0.0326x_6x_1 + 0.1254x_5x_3 - 0.1362x_5x_4 + 0.1664x_5x_2 - 0.4223x_5x_1 + 0.1039x_3x_4 - 0.0155x_3x_2 - 0.0735x_3x_1 - 0.1281x_4x_2 + 0.2183x_4x_1 - 0.2109x_2x_1$

Range $= 0.3576 - 0.0329x_6 + 0.1978x_5 + 0.0149x_3 - 0.0389x_4 - 0.4652x_2 + 0.4453x_1 + 0.0149x_6x_6 - 0.051x_5x_5 + 0.0075x_3x_3 - 0.0229x_4x_4 + 0.0987x_2x_2 - 0.0188x_1x_1 - 0.0524x_6x_5 - 0.0272x_6x_3 + 0.0281x_6x_4 - 0.0147x_6x_2 + 0.0083x_6x_1 + 0.1018x_5x_3 + 0.0563x_5x_4 - 0.0349x_5x_2 + 0.064x_5x_1 + 0.0073x_3x_4 + 0.0176x_3x_2 + 0.0341x_3x_1 + 0.1063x_4x_2 - 0.0374x_4x_1 + 0.0143x_2x_1$

Takeoff Field Length $= 0.2884 - 0.2896x_6 + 0.3376x_5 + 0.0088x_3 - 0.0478x_4 - 0.1448x_2 + 0.1239x_1 + 0.0714x_6x_6 - 0.029x_5x_5 + 0.0148x_3x_3 + 0.0068x_4x_4 + 0.2251x_2x_2 + 0.1654x_1x_1 - 0.12x_6x_5 - 0.0475x_6x_3 + 0.0426x_6x_4 - 0.0486x_6x_2 - 0.1058x_6x_1 + 0.1712x_5x_3 + 0.0071x_5x_4 - 0.0887x_5x_2 + 0.0759x_5x_1 + 0.0028x_3x_4 - 0.0056x_3x_2 + 0.064x_3x_1 + 0.0063x_4x_2 + 0.0456x_4x_1 - 0.2902x_2x_1$

Buffet Altitude $= 0.617 - 0.1221x_5 - 0.0485x_3 + 0.0141x_4 - 0.4507x_2 + 0.6968x_1 + 0.0248x_5x_5 + 0.0277x_3x_3 + 0.011x_4x_4 - 0.0873x_2x_2 - 0.295x_1x_1 - 0.061x_5x_3 - 0.0789x_5x_4 + 0.0546x_5x_2 - 0.1674x_5x_1 - 0.0008x_3x_4 + 0.0422x_3x_2 - 0.0371x_3x_1 + 0.017x_4x_2 - 0.0507x_4x_1 + 0.2845x_2x_1$

3.1.3 Vehicle Configuration Design Problem

The final problem used in this study is a vehicle configuration model (VCM) that was developed to evaluate the technical feasibility of new vehicle concepts [23,24,25]. Table

3 summarizes the problem definition that is used for this example. The inputs to the model are eleven high-level vehicle design parameters: ten continuous variables that define overall exterior dimensions and positions of the occupants, and one discrete variable, H, that defines the vehicle's powertrain as being one of six options: [1,2,3,4,5,6]. There are seven outputs from the model, including five measures of performance, vehicle mass, and total constraint violation (ConVio), which is zero when all of the constraints internal to the model are satisfied (i.e., ConVio = 0). The continuous design variables are normalized to [0,1] based on the input bounds while the objectives and vehicle mass are scaled against the baseline model. As noted in the table, it is desirable for Obj1 to be smaller than the baseline value while larger values are better for the other four objectives.

Table 3. VCM Problem Definition [4]

Model Inputs		
Variable	**Lower Bound**	**Upper Bound**
A	0	1
B	0	1
C	0	1
D	0	1
E	0	1
F	0	1
G	0	1
H	1,2,3,4,5, or 6	
I	0	1
J	0	1
K	0	1
Model Outputs		
ConVio	0 → feasible	> 0 → infeasible
Mass	Baseline = 1	Defines weight class
Obj1	Baseline = 1	Smaller is better
Obj2	Baseline = 1	Larger is better
Obj3	Baseline = 1	Larger is better
Obj4	Baseline = 1	Larger is better
Obj5	Baseline = 1	Larger is better

3.2 Description of User Trials

Two sets of user trials are defined for each of the three test problems based on an allocated number of function evaluations (~5,000 and ~10,000) that could be used, and each set of trials is run three times to account for any randomness in the algorithms, placement of attractors, or specification of brush/preferences controls. The function evaluation limits of 5,000 and 10,000 were set to allow for significant exploration of the trade space, yet they do not allow for an exhaustive search that would likely yield the entire Pareto frontier for each problem in every trial, as this is not the goal of the study. While there are nearly an infinite number of combinations of brushing, preference controls, and visual steering commands that could be implemented in ATSV, the trials are defined by stepping through a process that feels "natural" to create the first version (v1) of a trial and then those steps are replicated as accurately as possible to generate the second (v2) and third version (v3) of that trial. The first version of each trial is created by a user with a moderate experience level using ATSV and moderate familiarity with the test problems. The second and third versions are generated by users with a beginner level of experience using ATSV and limited familiarity with the test problems. This is done multiple times for each problem, creating five different combinations (Trials 1-5) that each used approximately 5,000 function evaluations and five different combinations (Trials 6-10) that each used approximately 10,000 function evaluations.

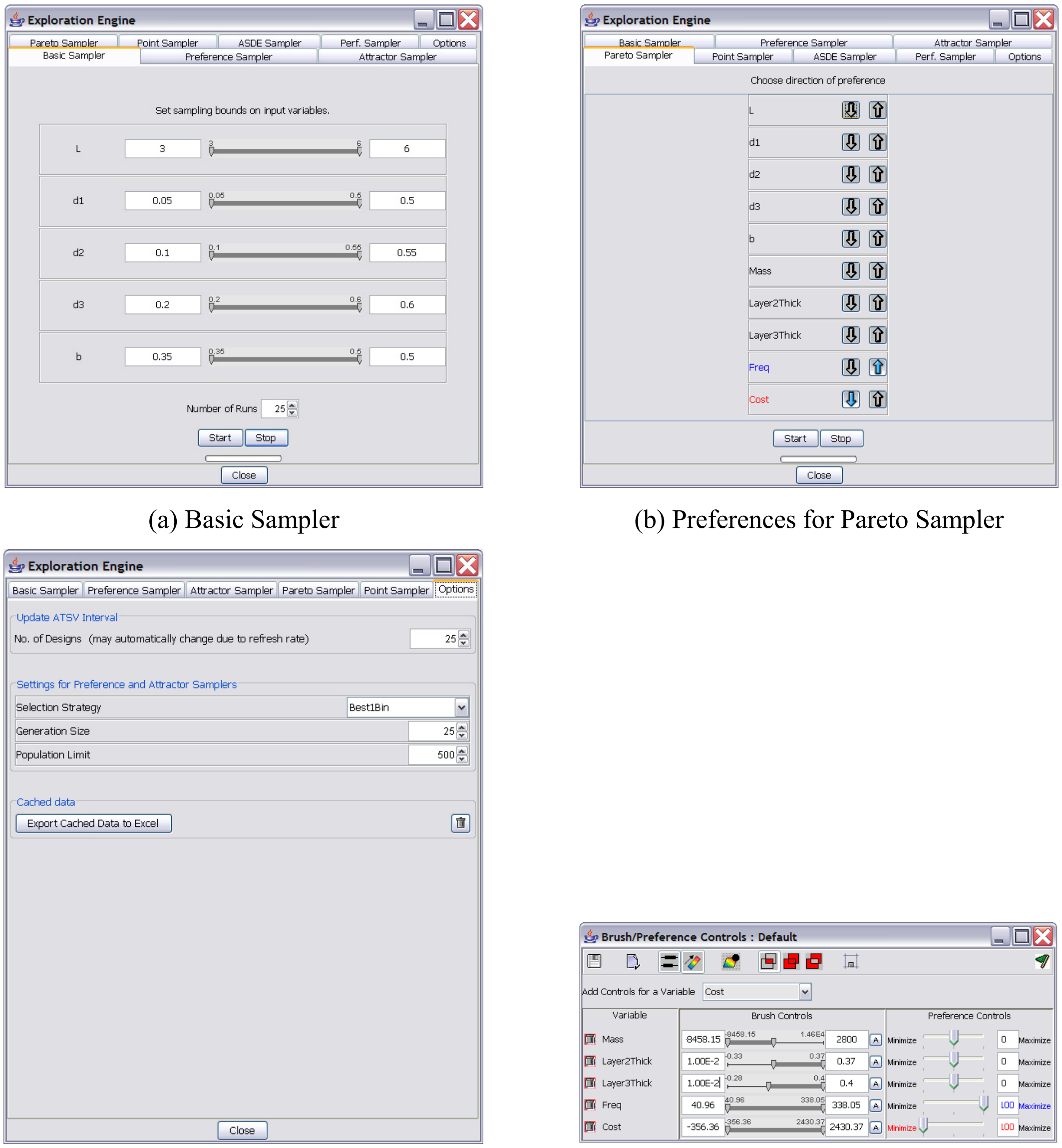
For the first two test problems (sandwich beam and aircraft wing), one trial in each group (5,000 and 10,000 function evaluations) is dedicated to running solely the Pareto sampler and one for running only the basic sampler over the entire input space. This is done to allow for comparison of the remaining trials to an automated search in showing how quickly the Pareto frontier is achieved. The vehicle configuration problem also contains a trial in each group that uses only the basic sampler, but it does not contain one that uses only the Pareto sampler; the user trials are instead compared to a multi-objective genetic algorithm [23,24,25].

Most trials begin with a relatively small set of randomly generated samples before proceeding to different combinations of samplers. This is done to give some perspective of the size and shape of the trade space as well as making some initial observations about relationships and tradeoffs. Initial motivation for setting some of the attractors comes from the fact that many designers use pair-wise comparisons in making decisions [27]; comparing only two objectives makes it easy to see relationships among them. However, not all attractors are placed for this reason; others are placed in an attempt to fill in gaps (similar to how the Gap Analyzer was used [25]) in the Pareto frontier, or to push the frontier toward optimality as the trade space exploration process unfolds. The preference and Pareto samplers are also used in an attempt to fill in the Pareto frontier. Specific details for each problem follow.

3.2.1 Settings for Sandwich Beam Design Problem

The ATSV set-up and parameter settings for all trials are shown in Figure 8. Table 4 and Table 5 (trials are intentionally out of order so that the table fits on the page) describe the specific combination of visual steering commands and brush/preference controls used for Trials 1-5 and Trials 6-10, respectively. Unless otherwise specified in the trial description, default option values are used (see Figure 8c of generation size = 25, population limit = 500, and the Best1Bin selection strategy).

Observations made about the trade space quickly reveal that points approaching values near ($L = 3$, $b = 0.35$, Layer2Thick ($d_2 - d_1$) = 0.01, and Layer3Thick ($d_3 - d_2$) = 0.01) dominated all other points in the feasible region. For this reason, many attractors contain some or all of these values. In addition, this is the motivation for Trials 5 and 10, which sample specifically near those designs. By sampling in this manner, all points generated contain ($L = 3$, $b = 0.35$) and have a much better chance of achieving (Layer2Thick = 0.01, and Layer3Thick = 0.01), especially Trial 10. Figure 9 shows the Pareto frontiers obtained by all three versions of Trial 3 while Figure 10 shows the Pareto frontiers obtained by all three versions of Trial 10, showing how all three versions of a trial result in similar Pareto frontiers.



(a) Basic Sampler

(b) Preferences for Pareto Sampler

(c) Sampler Options

(d) Preference Control Settings

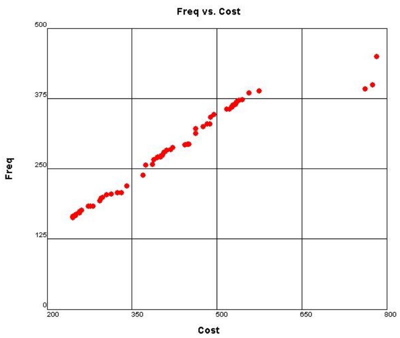
Figure 8. ATSV Set-up for the Sandwich Beam Design Problem Trials

Table 4. Specification of Visual Steering Commands for the Sandwich Beam Design Problem Trials 1-5

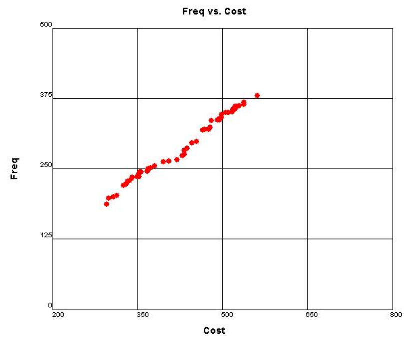
<table>
<tr><th>Trial 1 (Total Points: 5,050)</th><th>Trial 2 (Total Points: 5,050)</th></tr>
<tr><td>- Basic sampler: 250 runs
- Generation size changed to 50 for everything
- Brush and preference controls: Maximize (100) Freq, Minimize (-100) Cost, Mass ≤ 2800, Layer2Thick ≥ 0.01, Layer3Thick ≥ 0.01
- Preference sampler
- Brush and preference controls: Minimize (-100) L, Minimize (-100) b
- Preference sampler
- Brush and preference controls: L (0), B (0)
- Point attractors:
- [Cost = 562.028, Freq = 318.945]
- [Cost = 477.137, Freq = 312.950, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [L = 3.000, b = 0.350, Mass = 400, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [L = 3.000, b = 0.350, Mass = 1,016.703]
- [L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- Point attractor: Population limit changed to 1,000
- [L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [L = 3.000, b = 0.350]</td><td>- Preference controls:
- Maximize (100) Freq
- Minimize (-100) Cost
- Pareto sampler: Generation size changed to 50, population limit changed to 5,000
- Brush and preference controls:
- Mass ≤ 2800
- Layer2Thick ≥ 0.01
- Layer3Thick ≥ 0.01</td></tr>
<tr><th>Trial 3 (Total Points: 5,425)</th><th>Trial 4 (Total Points: 5,000)</th></tr>
<tr><td>- Basic sampler: 500 runs
- Generation size changed to 50 for everything
- Brush and preference controls: Maximize (100) Freq, Minimize (-100) Cost, Mass ≤ 2800, Layer2Thick ≥ 0.01, Layer3Thick ≥ 0.01
- Pareto sampler
- Point attractors: Population limit changed to 1000
- [L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 507.753, Freq = 353.717, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 560.636, Freq = 429.257, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 243.340, Freq = 155.875, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 630.219, Freq = 453.237, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- Point attractor:
- [Cost = 662.227, Freq = 466.427, L = 3.000, b = 0.350]</td><td>- Basic sampler: 5000 runs
- Brush and preference controls:
- Mass ≤ 2800
- Layer2Thick ≥ 0.01
- Layer3Thick ≥ 0.01</td></tr>
<tr><th colspan="2">Trial 5 (Total Points: 5,000)</th></tr>
<tr><td colspan="2">- Basic sampler: 500 runs
- Basic sampler: 500 runs (9x)
- L: 3.0 – 3.0 (each run)
- B: 0.35 – 0.35 (each run)
- d1: 0.05 – 0.10 (shifted up by 0.05 each run until upper limit is reached)
- d2: 0.10 – 0.15 (shifted up by 0.05 each run until upper limit is reached)
- d3: 0.20 – 0.25 (shifted up by 0.05 each run until upper limit is reached)
- Brush and preference controls:
- Maximize (100) Freq
- Minimize (-100) Cost
- Mass ≤ 2800
- Layer2Thick ≥ 0.01
- Layer3Thick ≥ 0.01</td></tr>
</table>

Table 5. Specification of Visual Steering Commands for the Sandwich Beam Design Problem Trials 6-10

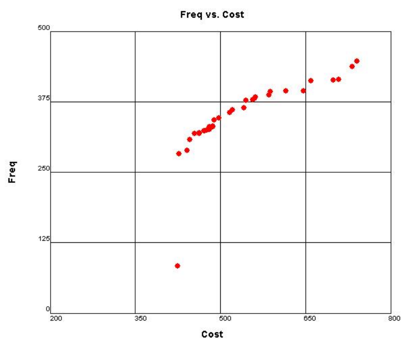
<table>
<tr><th>Trial 6 (Total Points: 10,849)</th><th>Trial 10 (Total Points: 10,000)</th></tr>
<tr><td>- Basic sampler: 250 runs
- Generation size changed to 50 for everything
- Brush and preference controls:
- Maximize (100) Freq
- Minimize (-100) Cost
- Mass ≤ 2800
- Layer2Thick ≥ 0.01
- Layer3Thick ≥ 0.01
- Point attractors: Population limit changed to 1000
- [Cost = 250, Freq = 185, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 346, Freq = 250, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 387, Freq = 280, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 422, Freq = 305, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 446, Freq = 335, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 472, Freq = 364, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 491, Freq = 382, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 526, Freq = 422, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- [Cost = 580, Freq = 448, L = 3.000, b = 0.350]
- [Cost = 625, Freq = 469, L = 3.000, b = 0.350, Layer2Thick = 0.030, Layer3Thick = 0.030]
- [Cost = 667, Freq = 469, L = 3.000, b = 0.350, Layer2Thick = 0.030, Layer3Thick = 0.030]
- [Cost = 687, Freq = 477, L = 3.000, b = 0.350, Layer2Thick = 0.030, Layer3Thick = 0.030]
- [Cost = 695, Freq = 471, L = 3.000, b = 0.350]</td>
<td>- Basic sampler: 1500 runs
- Basic sampler:
- L: 3.0 – 3.0 (every run)
- B: 0.35 – 0.35 (every run)
- Basic sampler: 250 runs
- d1: 0.05 – 0.08 (shifted up by 0.03 after each run)
- d2: 0.10 – 0.13 (0.11 – 0.15 on 2nd run, shifted up by 0.03 after each run, 0.23 – 0.29 on 7th run, shifted up by 0.03 after each run)
- d3: 0.20 – 0.23 (0.20 – 0.25 on 6th run, 0.23 – 0.30 on 7th run, shifted up by 0.03 after each run)
- Basic sampler: 500 runs
- d1: 0.26 – 0.29 (shifted up by 0.03 after each run until upper limit is reached)
- d2: 0.26 – 0.31 (shifted up by 0.03 after each run until upper limit is reached)
- d3: 0.26 – 0.33 (shifted up by 0.03 after each run until upper limit is reached)
- Basic sampler: 1000 runs
- d1: 0.47 – 0.50
- d2: 0.47 – 0.55
- d3: 0.55 – 0.60
- Basic sampler: 750 runs
- d1: 0.20 – 0.23
- d2: 0.20 – 0.25
- d3: 0.20 – 0.27
- Brush and preference controls: Maximize (100) Freq, Minimize (-100) Cost, Mass ≤ 2800, Layer2Thick ≥ 0.01, Layer3Thick ≥ 0.01</td></tr>
<tr><th>Trial 8 (Total Points: 10,850)</th><th>Trial 9 (Total Points: 10,000)</th></tr>
<tr><td>- Basic sampler: 500 runs
- Generation size changed to 50 for everything
- Brush and preference controls: Maximize (100) Freq, Minimize (-100) Cost, Mass ≤ 2800, Layer2Thick ≥ 0.01, Layer3Thick ≥ 0.01
- Point attractors: Population limit changed to 1000
- [L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010] (10X)
- [Cost = 343, Freq = 262, L = 3.000, b = 0.350, Layer2Thick = 0.010, Layer3Thick = 0.010]
- Basic sampler: 1000 runs
- L: 3.0 – 3.0
- B: 0.35 – 0.35
- d1: 0.47 – 0.50
- d2: 0.47 – 0.55
- d3: 0.55 – 0.60</td>
<td>- Basic sampler: 10000 runs
- Brush and preference controls:
- Mass ≤ 2800
- Layer2Thick ≥ 0.01
- Layer3Thick ≥ 0.01</td></tr>
<tr><th colspan="2">Trial 7 (Total Points: 10,050)</th></tr>
<tr><td colspan="2">- Preference controls: Maximize (100) Freq, Minimize (-100) Cost
- Pareto sampler: Generation size changed to 50, population limit changed to 10,000
- Brush and preference controls: Mass ≤ 2800, Layer2Thick ≥ 0.01, Layer3Thick ≥ 0.01</td></tr>
</table>



(a) Trial 3 v1 Pareto Frontier

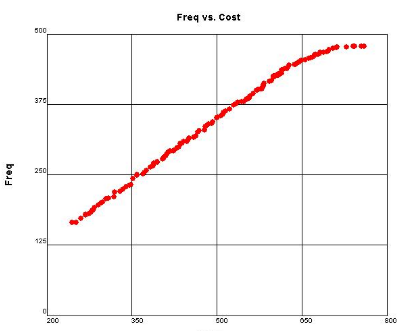


(b) Trial 3 v2 Pareto Frontier

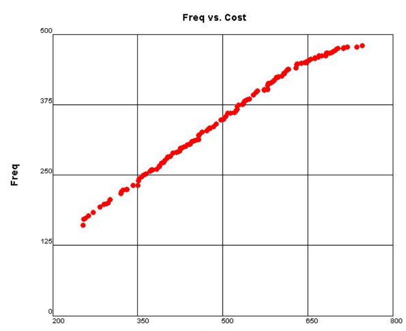


(c) Trial 3 v3 Pareto Frontier

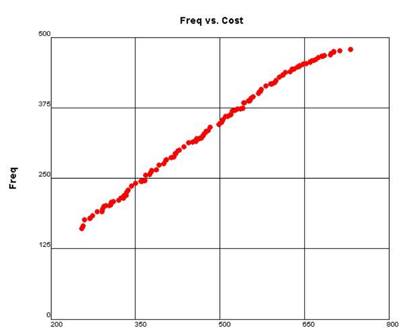
Figure 9. Example of Results, Sandwich Beam Design Problem Trial 3



(a) Trial 10 v1 Pareto Frontier



(b) Trial 10 v2 Pareto Frontier

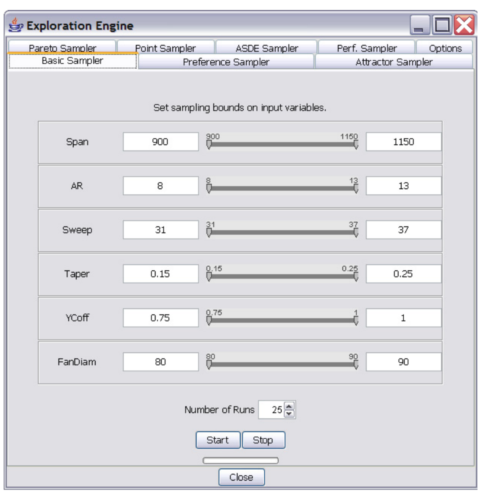


(c) Trial 10 v3 Pareto Frontier

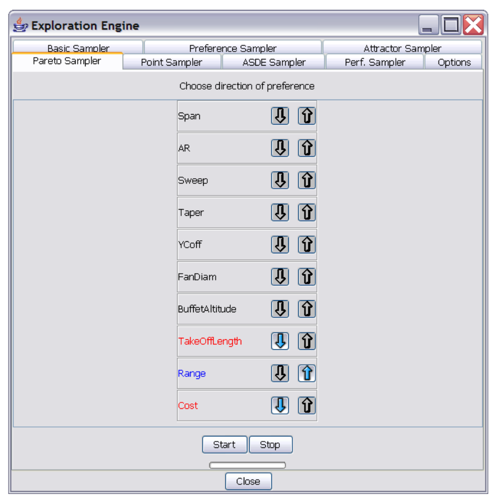
Figure 10. Example of Results, Sandwich Beam Design Problem Trial 10

3.2.2 Settings for Aircraft Wing Design Problem

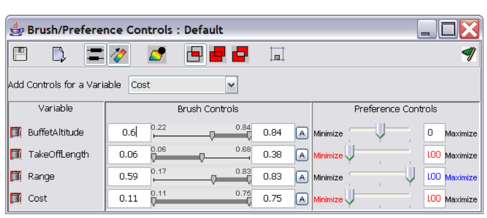
The ATSV set-up and parameter settings for all trials for the aircraft wing design problem are shown in Figure 11. Table 6 and Table 7 describe the specific combination of visual steering commands and brush/preference controls used for Trials 1-5 and Trials 6-10, respectively. Like the sandwich beam problem, unless otherwise specified in the trial description, settings are left at the default values seen in Figure 8c. In these trials, most attractors are placed with the goal of achieving feasible points, pulling the frontier further out, and filling gaps in the frontier. Unlike the sandwich beam problem, the Pareto points of this problem varied widely in their input values and the same basic sampling strategy could not be used. Figure 12 shows the Pareto frontiers obtained by all three versions of Trial 5 while Figure 13 shows the Pareto frontiers obtained by all three versions of Trial 6, again, showing that all three versions of a trial result in similar Pareto frontiers.



(a) Basic Sampler



(b) Preferences for Pareto Sampler



(c) Preference Control Settings

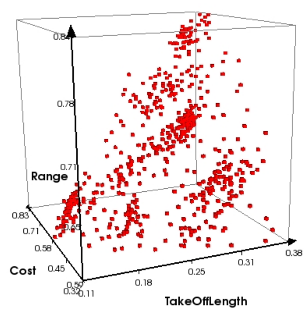
Figure 11. ATSV Set-up for the Aircraft Wing Design Problem Trials

Table 6. Specification of Visual Steering Commands for the Aircraft Wing Design Problem Trials 1-5

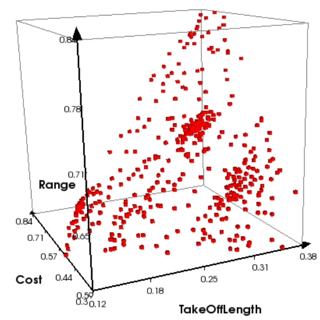
<table>
<tr><th>Trial 1 (Total Points: 5,379)</th><th>Trial 2 (Total Points: 5,100)</th></tr>
<tr><td>- Basic sampler: 100 runs
- Generation size changed to 60 for everything
- Brush and preference controls: Maximize (100) Range ≥ 0.589, BuffetAltitude (50) ≥ 0.603, Minimize (-33) TOFL ≤ 0.377, Minimize (-33) Cost
- Preference Sampler: Population limit changed to 600
- Brush and preference controls: BuffetAltitude (0), TOFL (-100), Cost (-100)
- Point attractors: Population limit changed to 600
- [Cost = 0.582, Range = 0.804, TOFL = 0.243]
- [Cost = 0.439, Range = 0.770, TOFL = 0.319]
- [Cost = 0.631, Range = 0.746, TOFL = 0.190]
- [Cost = 0.690, Range = 0.825, TOFL = 0.263]
- [Cost = 0.571, Range = 0.823, TOFL = 0.323]
- [Cost = 0.512, Range = 0.747, TOFL = 0.210]
- [Cost = 0.524, Range = 0.710, TOFL = 0.162]
- [Cost = 0.363, Range = 0.697, TOFL = 0.307]</td><td>- Preference controls:
- Maximize Range
- Minimize TOFL
- Minimize Cost
- Pareto sampler: Generation size changed to 60, population limit changed to 5,000
- Brush and preference controls:
- Range ≥ 0.589
- BuffetAltitude ≥ 0.603
- TOFL ≤ 0.377</td></tr>
<tr><th>Trial 3 (Total Points: 5,109)</th><th>Trial 4 (Total Points: 5,000)</th></tr>
<tr><td>- Basic sampler: 250 runs
- Generation size changed to 60 for everything
- Brush and preference controls: Maximize (100) Range ≥ 0.589, BuffetAltitude ≥ 0.603, Minimize (-100) TOFL ≤ 0.377, Minimize (-100) Cost
- Pareto sampler: Population limit changed to 600
- Point attractors: Population limit changed to 600
- [Span = 1150, Cost = 0.578, Range = 0.832, TOFL = 0.227, BuffetAltitude = 0.810]
- [Span = 1150, Cost = 0.435, Range = 0.779, TOFL = 0.278, BuffetAltitude = 0.620]
- [Span = 1150, Cost = 0.687, Range = 0.791, TOFL = 0.197, BuffetAltitude = 0.925]
- [Span = 1150, Cost = 0.510, Range = 0.719, TOFL = 0.190, BuffetAltitude = 0.805]
- [Span = 1150, Cost = 0.532, Range = 0.764, TOFL = 0.267, BuffetAltitude = 0.805]
- [Span = 1150, Cost = 0.741, Range = 0.837, TOFL = 0.305, BuffetAltitude = 0.962]
- [Span = 1150, Cost = 0.537, Range = 0.625, TOFL = 0.102, BuffetAltitude = 0.916]</td><td>- Basic sampler: 5000 runs
- Brush and preference controls:
- Maximize (100) Range ≥ 0.589
- BuffetAltitude ≥ 0.603
- Minimize (-100) TOFL ≤ 0.377
- Minimize (-100) Cost</td></tr>
<tr><th colspan="2">Trial 5 (Total Points: 5,290)</th></tr>
<tr><td colspan="2">- Basic sampler: 250 runs
- Generation size changed to 60 for everything
- Brush and preference controls: Maximize (100) Range ≥ 0.589, BuffetAltitude ≥ 0.603, Minimize (-100) TOFL ≤ 0.377, Minimize (-100) Cost
- Pareto sampler: Population limit changed to 600
- Point attractors: Population limit changed to 600
- [Cost = 0.715, Range = 0.853, TOFL = 0.321, BuffetAltitude = 0.908]
- [Cost = 0.364, Range = 0.719, TOFL = 0.287, BuffetAltitude = 0.700]
- Point attractors: Population limit changed to 1000
- [Cost = 0.556, Range = 0.674, TOFL = 0.102, BuffetAltitude = 0.879]
- [Cost = 0.488, Range = 0.763, TOFL = 0.270, BuffetAltitude = 0.785]
- Point attractors: Population limit changed to 600
- [Cost = 0.624, Range = 0.776, TOFL = 0.270, BuffetAltitude = 0.928]
- [Cost = 0.437, Range = 0.688, TOFL = 0.187, BuffetAltitude = 0.802]</td></tr>
</table>

Table 7. Specification of Visual Steering Commands for the Aircraft Wing Design Problem Trials 6-10

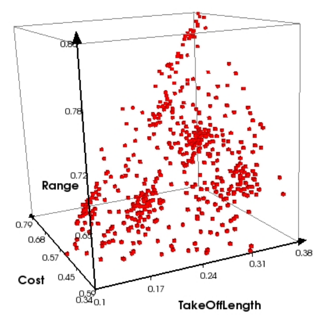
<table>
<tr><th>Trial 6 (Total Points: 10,345)</th><th>Trial 7 (Total Points: 10,105)</th></tr>
<tr><td>- Basic sampler: 100 runs
- Generation size changed to 60 and Population limit changed to 1000 for everything
- Brush and preference controls: Maximize (100) Range ≥ 0.589, BuffetAltitude ≥ 0.603, Minimize (-100) TOFL ≤ 0.377, Minimize (-100) Cost
- Point attractors:
- [Cost = 0.511, Range = 0.765, TOFL = 0.290]
- [Cost = 0.657, Range = 0.786, TOFL = 0.217]
- [Cost = 0.566, Range = 0.753, TOFL = 0.213]
- [Cost = 0.420, Range = 0.703, TOFL = 0.255]
- [Cost = 0.707, Range = 0.721, TOFL = 0.149]
- [Cost = 0.627, Range = 0.823, TOFL = 0.289]
- [Cost = 0.403, Range = 0.724, TOFL = 0.312]
- [Cost = 0.480, Range = 0.746, TOFL = 0.255]
- [Cost = 0.587, Range = 0.816, TOFL = 0.315]
- [Cost = 0.525, Range = 0.704, TOFL = 0.180]
- [Cost = 0.707, Range = 0.840, TOFL = 0.313]</td><td>- Preference controls:
- Maximize Range
- Minimize TOFL
- Minimize Cost
- Pareto sampler: Generation size changed to 60, population limit changed to 10,000
- Brush and preference controls:
- Range ≥ 0.589
- BuffetAltitude ≥ 0.603
- TOFL ≤ 0.377</td></tr>
<tr><th>Trial 8 (Total Points: 10,520)</th><th>Trial 9 (Total Points: 10,000)</th></tr>
<tr><td>- Basic sampler: 500 runs
- Generation size changed to 60 for everything
- Brush and preference controls: Maximize (100) Range ≥ 0.589, BuffetAltitude (50) ≥ 0.603, Minimize (-33) TOFL ≤ 0.377, Minimize (-33) Cost
- Preference Sampler: Population limit changed to 1000
- Brush and preference controls: BuffetAltitude (0), TOFL (-100), Cost (-100)
- Pareto sampler: Population limit changed to 1000
- Point attractors: Population limit changed to 1000
- [Cost = 0.726, Range = 0.886, TOFL = 0.316, BuffetAltitude = 0.900]
- [Cost = 0.598, Range = 0.689, TOFL = 0.115, BuffetAltitude = 0.900]
- [Cost = 0.701, Range = 0.770, TOFL = 0.177, BuffetAltitude = 0.922]
- [Cost = 0.450, Range = 0.736, TOFL = 0.253, BuffetAltitude = 0.713]
- [Cost = 0.559, Range = 0.798, TOFL = 0.278, BuffetAltitude = 0.803]
- [Cost = 0.546, Range = 0.716, TOFL = 0.184, BuffetAltitude = 0.788]
- [Cost = 0.372, Range = 0.709, TOFL = 0.320, BuffetAltitude = 0.610]
- [Cost = 0.475, Range = 0.764, TOFL = 0.330, BuffetAltitude = 0.610]
- [Cost = 0.591, Range = 0.839, TOFL = 0.333, BuffetAltitude = 0.806]
- [Cost = 0.520, Range = 0.784, TOFL = 0.282, BuffetAltitude = 0.806]</td><td>- Basic sampler: 10000 runs
- Brush and preference controls:
- Maximize (100) Range ≥ 0.589
- BuffetAltitude ≥ 0.603
- Minimize (-100) TOFL ≤ 0.377
- Minimize (-100) Cost</td></tr>
<tr><th colspan="2">Trial 10 (Total Points: 10,340)</th></tr>
<tr><td colspan="2">- Basic sampler: 500 runs
- Generation size changed to 60 for everything
- Brush and preference controls: Maximize (100) Range ≥ 0.589, BuffetAltitude ≥ 0.603, Minimize (-100) TOFL ≤ 0.377, Minimize (-100) Cost
- Pareto sampler: Population limit changed to 1200
- Point attractors: Population limit changed to 1200
- [Cost = 0.727, Range = 0.883, TOFL = 0.327]
- [Cost = 0.545, Range = 0.649, TOFL = 0.111]
- Pareto sampler: Population limit changed to 1200
- Point attractors: Population limit changed to 1200
- [Cost = 0.385, Range = 0.749, TOFL = 0.341]
- [Cost = 0.440, Range = 0.722, TOFL = 0.232]
- Pareto sampler: Population limit changed to 1200
- Point attractors: Population limit changed to 1200
- [Cost = 0.605, Range = 0.758, TOFL = 0.207]
- [Cost = 0.622, Range = 0.847, TOFL = 0.304]</td></tr>
</table>



(a) Trial 5 v1 Pareto Frontier

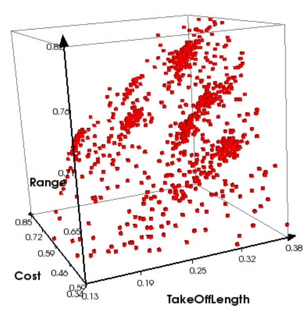


(b) Trial 5 v2 Pareto Frontier

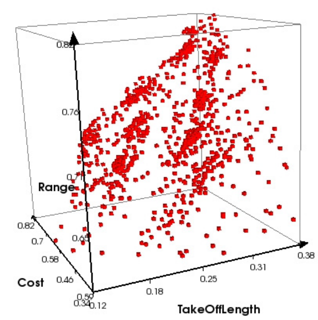


(c) Trial 5 v3 Pareto Frontier

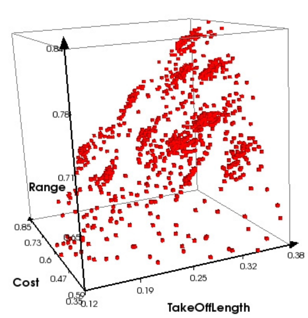
Figure 12. Example of Results, Aircraft Wing Design Problem Trial 5



(a) Trial 6 v1 Pareto Frontier



(b) Trial 6 v2 Pareto Frontier

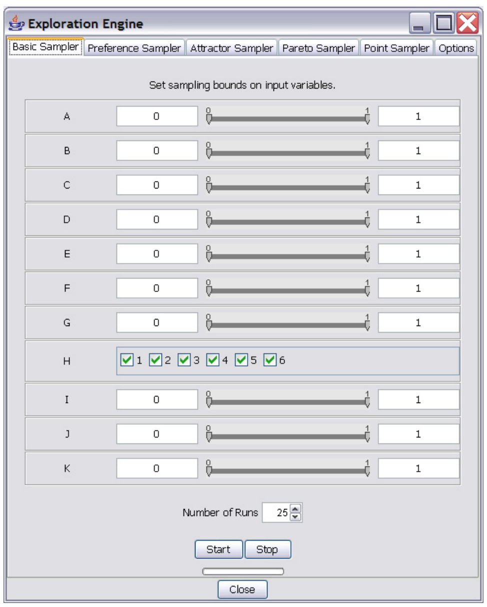


(c) Trial 6 v3 Pareto Frontier

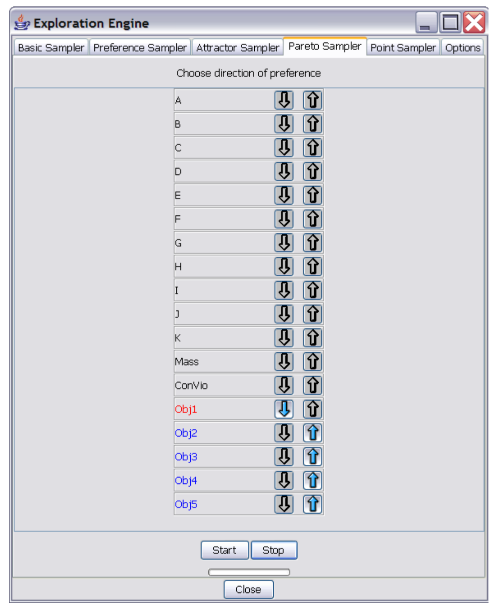
Figure 13. Example of Results, Aircraft Wing Design Problem Trial 6

3.2.3 Settings for Vehicle Configuration Design Problem

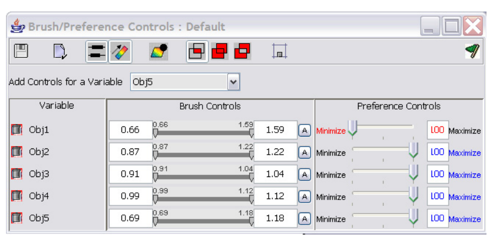
The ATSV set-up and parameter settings for Trials 1-5 for the vehicle configuration model are shown in Figure 14. Table 8 describes the specific combination of visual steering commands and brush/preference controls used for Trials 1-5. As with the other test problems, unless otherwise specified, the default option values are used (see Figure 8c). Figure 15 shows the Pareto frontiers obtained by all three versions of Trial 4.



(a) Basic Sampler

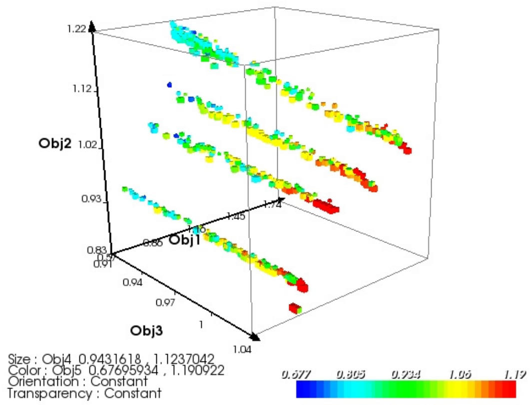


(b) Preferences for Pareto Sampler

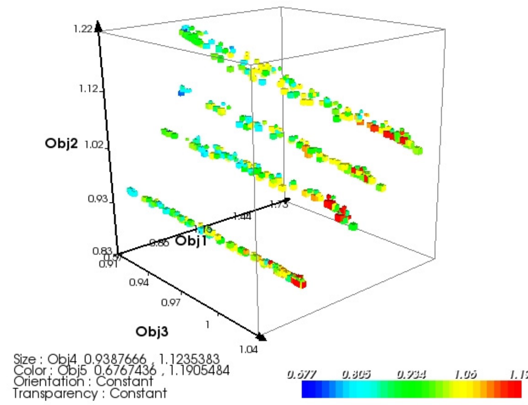


(c) Preference Control Settings

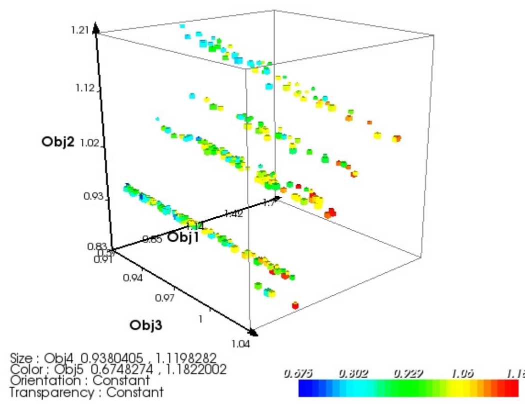
Figure 14. ATSV Set-up for VCM Trials 1-5



(a) Trial 4 v1 Pareto Frontier



(b) Trial 4 v2 Pareto Frontier



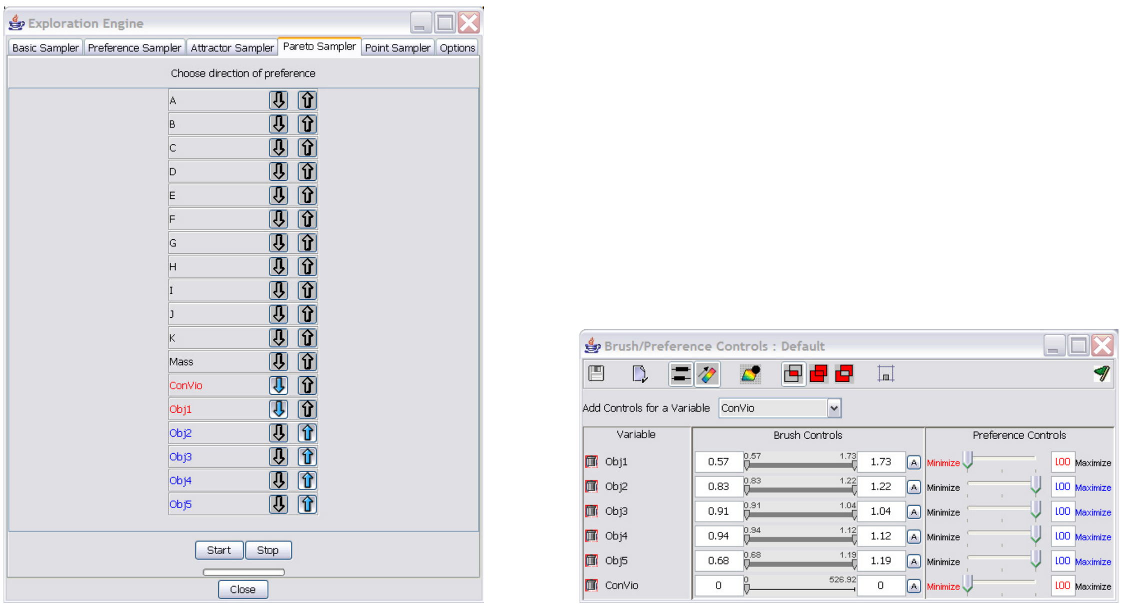
(c) Trial 4 v3 Pareto Frontier

Figure 15. Example of Results, VCM Trial 4

Table 8. Specification of Visual Steering Commands for VCM Trials 1-5

<table>
<tr><th>Trial 1 (Total Points: 5,025)</th><th>Trial 2 (Total Points: 5,075)</th></tr>
<tr><td>- Basic Sampler: 100 runs
- Brush objectives 1-5: Minimize objective 1 (-100), maximize objectives 2-5 (100)
- Point attractors: 10 possible pair-wise point attractors for objectives 1-5 set at the current limits of the scatter plot window (on objectives [1 & 2], [3 & 4], [5 & 1], [2 & 3], [4 & 5], [1 & 3], [2 & 4], [3 & 5], [4 & 1], [5 & 2])
- Pareto Sampler</td><td>- Basic Sampler: 500 runs
- Brush objectives 1-5: Minimize objective 1 (-100), maximize objectives 2-5 (100)
- Pareto Sampler
- Line attractors (1-d point attractor): One for each objective 1-5 set at the current limit of the scatter plot window (minimum of window for objective 1 and maximum of window for objectives 2-5)
- Preference Sampler
- Point attractors: Set at the current limits of the scatter plot window (on objectives [2 & 5], [2 & 4])
- Point attractors: Set at the current limits of the scatter plot window, generation size changed to 15 (on objectives [3 & 2], [3 & 4], [1 & 5], [2 & 5])
- Point attractor: Set at the current limits of the scatter plot window (on objectives [3 & 5])</td></tr>
<tr><th>Trial 3 (Total Points: 5,525)</th><th>Trial 4 (Total Points: 5,375)</th></tr>
<tr><td>- Basic Sampler: 500 runs
- Brush objectives 1-5: Minimize objective 1 (-100), maximize objectives 2-5 (100)
- Point attractors: Set at the current limits of the glyph plot window (on objectives [1, 2, & 3], [1, 2, & 4], [1, 2, & 5], [1, 3, & 4], [1, 3, & 5], [1, 4, & 5], [2, 3, & 4], [2, 3, & 5], [2, 4, & 5], [3, 4, & 5])
- Pareto Sampler</td><td>- Basic Sampler: 100 runs
- Brush objectives 1-5: Minimize objective 1 (-100), maximize objectives 2-5 (100)
- Line attractors (1-d point attractors): Set at the current limits of the scatter plot window (on objectives 1–5)
- Pareto Sampler
- Point attractors: Set at the current limits of the scatter plot window, generation size changed to 15 and population limit changed to 250 (on objectives [1 & 2], [1 & 3], [1 & 4], [1 & 5], [2 & 3], [2 & 4], [2 & 5], [3 & 4], [3 & 5], [4 & 5])
- Line attractor (1-d point attractor): On objective 3 set at the current limit of the scatter plot window
- Point attractors: Set at the current limits of the scatter plot window (on objectives [3 & 4], [4 & 5])</td></tr>
<tr><th colspan="2">Trial 5 (Total Points: 5,000)</th></tr>
<tr><td colspan="2">- Basic Sampler: 5,000 runs
- Brush objectives 1-5: Minimize objective 1 (-100), maximize objectives 2-5 (100)</td></tr>
</table>

The second set of five trials (Trials 6-10) for the vehicle configuration model use approximately 10,000 points each, doubling the number of function evaluations allocated. These trials, with the exception of Trial 10, all begin with a small set of random samples to allow the user to specify preferences (see Figure 16), but they then vary widely in the order and type of attractors and samplers used. Table 9 describes the preference settings and specific combination of visual steering commands that are used for Trials 6-10. Note that these trials also set a preference on ConVio to minimize it before generating too many points, with the exception of Trial 5, which sets it halfway through the trial. Unless specified, the same options and parameter settings are used for these trials as Trials 1-5 (see Figure 8c). Figure 17 shows an example of the Pareto frontiers from all three versions of Trial 6.



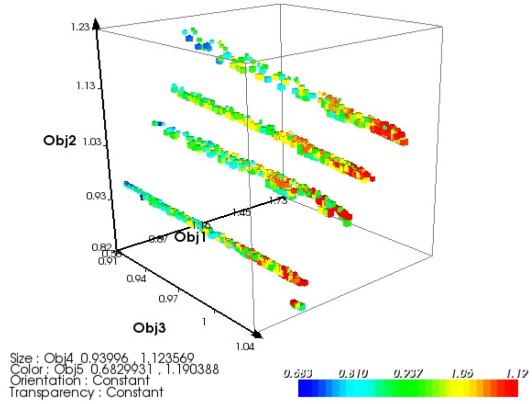
(a) Preferences for Pareto Sampler

(b) Preference Control Settings

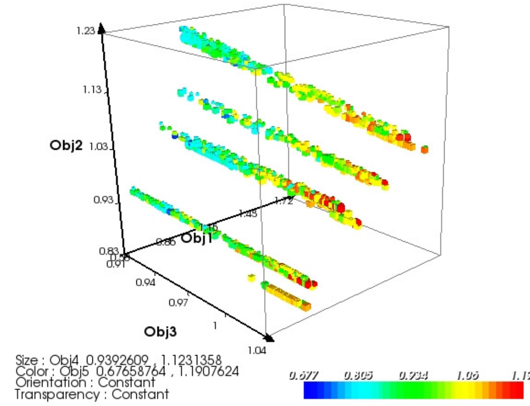
Figure 16. ATSV Set-up for VCM Trials 6-10

Table 9. Specification of Visual Steering Commands for VCM Trials 6-10

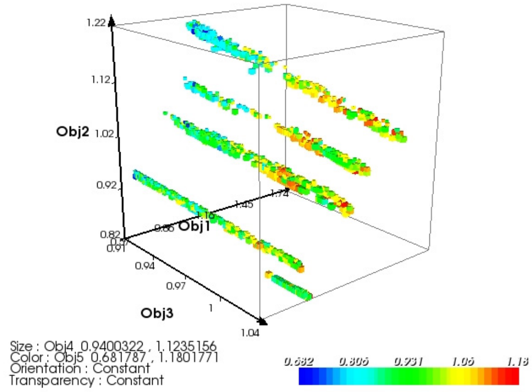
<table>
<tr><th>Trial 6 (Total Points: 10,325)</th><th>Trial 7 (Total Points: 10,075)</th></tr>
<tr><td>- Basic Sampler: 100 runs
- Brush objectives 1-5: Minimize objective 1 (-100), maximize objectives 2-5 (100)
- Point attractors: Set at the current limits of the scatter plot window ± 5% for minimizing or maximizing, respectively (on objectives [1 & 2], [2 & 3], [3 & 4], [4 & 5], [5 & 1], [1 & 3], [3 & 5], [5 & 2], [2 & 4], [4 & 1])
- Preference Sampler
- Point attractors: These specific values were used to fill in the Pareto frontier ([Obj1 = 0.9, Obj2 = 1.102], [Obj1 = 0.645, Obj2 = .872], [Obj2 = 1.144, Obj3 = .988])
- Line attractors (1-d point attractors): These specific values were used to fill in the Pareto frontier ([Obj4 = 1.124], [Obj5 = 1.191])
- Brush (preference): Minimize ConVio (-100)
- Preference Sampler
- Pareto Sampler
- Line attractors (1-d point attractors): One for each objective 1-5 set at the feasible limit of the objective in the scatter window (minimum for objective 1 and maximum for objectives 2-5)
- Pareto Sampler</td>
<td>- Basic Sampler: 250 runs
- Brush objectives 1-5 and ConVio: Minimize objective 1 and ConVio (-100), maximize objectives 2-5 (100)
- Preference Sampler: Generation size changed to 50 and population limit changed to 1,000
- Pareto Sampler: Generation size changed to 50 and population limit changed to 1,000
- Point attractors: Set at the current limits of the scatter plot window (on [ConVio & Obj1], [ConVio & Obj2], [ConVio & Obj3], [ConVio & Obj4], [ConVio & Obj5])
- Pareto Sampler Generation size changed to 50 and population limit changed to 1,000
- Point attractors: These specific values were used to fill in the Pareto frontier ([ConVio = 0, Obj1 = 1.043, Obj2 = 1.2], [ConVio = 0, Obj1 = .755, Obj3 = 1.026], [ConVio = 0, Obj1 = .911, Obj4 = 1.121], [ConVio = 0, Obj1 = .729, Obj2 = 1.153], [ConVio = 0, Obj2 = 1.126, Obj3 = .993], [ConVio = 0, Obj2 = 1.186, Obj4 = 1.099], [ConVio = 0, Obj2 = 1.154, Obj5 = 1.052], [ConVio = 0, Obj3 = 1.018, Obj4 = 1.123], [ConVio = 0, Obj3 = 1.003, Obj5 = 1.137], [ConVio = 0, Obj4 = 1.121, Obj5 = 1.105], [ConVio = 0, Obj3 = .923, Obj5 = .993], [ConVio = 0, Obj2 = 1.207, Obj5 = .853])
- Preference Sampler
- Pareto Sampler
- Point attractors: ([Obj1 = .802, Obj2 = .851, Obj3 = 1.007], [Obj3 = 1.003, Obj2 = .854], [Obj1 = 1.073, Obj2 = 1.19], [Obj4 = .995, Obj5 = .824], [Obj3 = .955, Obj4 = 1.119])
- Pareto Sampler: Population limit changed to 250</td></tr>
<tr><th>Trial 8 (Total Points: 10,125)</th><th>Trial 9 (Total Points: 10,275)</th></tr>
<tr><td>- Basic Sampler: 25 runs
- Brush objectives 1-5 and ConVio: Minimize objective 1 and ConVio (-100), maximize objectives 2-5 (100)
- Pareto Sampler: Generation size changed to 50 and population limit changed to 1,000
- Preference Sampler: Generation size changed to 50 and population limit changed to 1,000
- Repeated Pareto and Preference Samplers in above order with the same settings four more times
- Pareto Sampler: Generation size changed to 50 and population limit changed to 1,000</td>
<td>- Basic Sampler: 25 runs
- Brush objectives 1-5 and ConVio: Minimize objective 1 and ConVio (-100), maximize objectives 2-5 (100)
- Pareto Sampler: Generation size changed to 50, population limit changed to 1,000, and selection strategy changed to Rand1Bin
- Preference Sampler: Generation size changed to 50, population limit changed to 1,000, and selection strategy changed to Rand1Bin
- Repeated Pareto and Preference Samplers in above order with the same settings four more times</td></tr>
<tr><th colspan="2">Trial 10 (Total Points: 10,000)</th></tr>
<tr><td colspan="2">- Basic Sampler: 10,000 runs
- Brush objectives 1-5: Minimize objective 1 (-100), maximize objectives 2-5 (100)</td></tr>
</table>



(a) Trial 6 v1 Pareto Frontier



(b) Trial 6 v2 Pareto Frontier



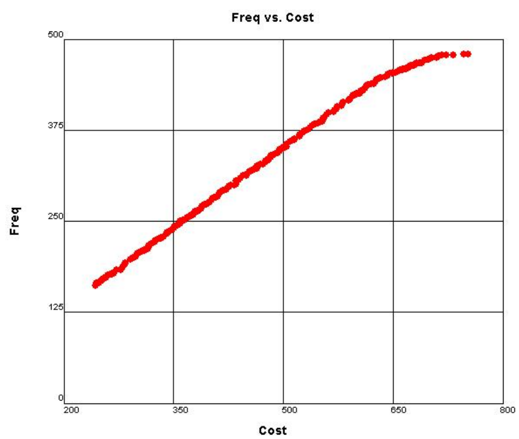
(c) Trial 6 v3 Pareto Frontier

Figure 17. Example of Results, VCM Trial 6

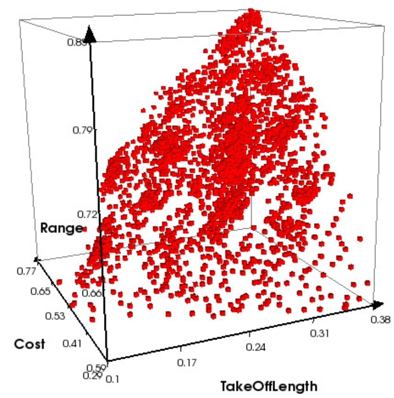
3.3 Reference Pareto Sets

For consistency, the reference (or "best known") Pareto frontier for each problem is generated by combining the Pareto fronts from all three versions of every trial in both sets of trials into a "super set" and then using ATSV to perform a non-dominated sort of this "super set." Figure 18 shows the reference Pareto sets for all three test problems.

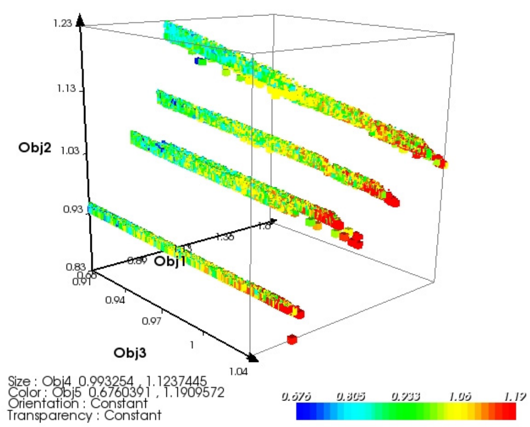
In addition, for the vehicle configuration design problem, three exhaustive runs of the multi-objective genetic algorithm (MOGA) [24] were performed to find the Pareto frontier. In order to ensure the Pareto frontier generated by the exhaustive MOGA contained no large holes or gaps (i.e., covered the entire objective space), a Gap Analyzer was used to direct the MOGA to find designs in those areas if such a region was found [25]. The exhaustive MOGA used approximately 80,000 function evaluations to create its Pareto frontier. Even with the Gap Analyzer, though, the MOGA ran “blindly”, requiring no human intervention while searching the trade space; hence, it provides a suitable benchmark for this study.



(a) Sandwich Beam



(b) Aircraft Wing



(c) VCM

Figure 18. Test Problem Reference Pareto Frontiers

3.4 Performance Metric

To quantify the performance and compare the results of multi-objective genetic algorithms rigorously, a variety of performance metrics have been developed [28]. Okabe, et al. [29] states that these metrics should be used to assess (1) the number of Pareto-optimal solutions in the set, (2) the closeness of the solutions to the theoretical Pareto-front, and (3) the distribution and spread of the solutions. Zitzler [30] proposed a hyper-volume metric, which evaluates the size of the dominated space. Reed and Tang [31] have developed and refined performance metrics to evaluate two Pareto frontiers in a 5-D trade space. In particular, ε-performance has been used to assess the performance of multi-objective genetic algorithms due to its relative computational efficiency, accuracy, and ease-of-use. Since only relative performances of the trials are needed, the ε-performance metric developed by Kollat and Reed [32,33] was selected as the basis for comparison.

This ε-performance metric assesses the proportion of solutions that were found within a user-specified level of precision relative to the "true" Pareto frontier, or best available reference set. In other words, the user can specify a precision level for each objective to tailor it to a given application. The solutions are then evaluated with respect to the reference set based on this user specified precision. The proportion (percentage) of solutions in the reference set that are found by the genetic algorithm within this level of precision is reported as its ε-performance. Since the solutions are evaluated with respect to a best known reference Pareto set, it is possible that the solutions may at times dominate reference set solutions. To account for this, ε-performance is reported as the proportion of reference set solutions that are dominated, or found within the user-specified ε precision. These metrics allow for numerical comparison between the solutions generated using the different combinations of visual steering commands within ATSV and the reference Pareto frontiers.

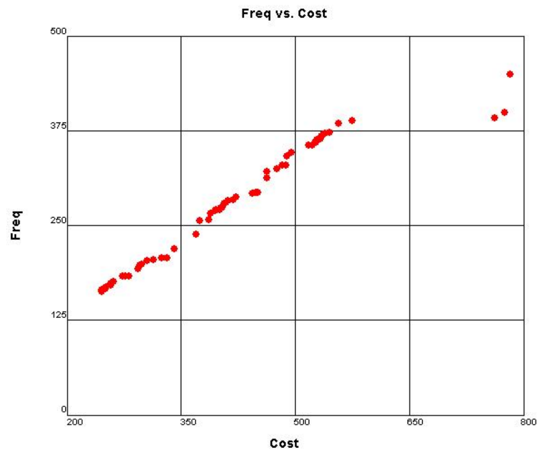
Before comparing the sets of solutions quantitatively using the ε-performance metric, a suitable value for epsilon needs to be determined. After confirming that all output variables are normalized by the same ranges (VCM and aircraft wing problem) the differences between the objectives of every pair of designs in the reference sets are computed. It is found that the smallest difference between any two designs is so close to zero that any reasonable value of epsilon could be selected. While choosing an epsilon value that is too large would reduce each set to the point that comparison would be meaningless, choosing an epsilon value that is too small would make it almost impossible to find designs within one epsilon of each other in each objective, especially in the VCM case given that it is a 5-D space. In addition, since the goal is to compare the relative performances of the different trials to one another, an epsilon value needs to be chosen that ensures that the reduced sets are still representative of the originals. Therefore, a value of 0.01 is selected for this analysis and used for each of the objectives (VCM and wing problem). The sandwich beam problem is slightly different in that its output values are not normalized to range between 0 and 1 or scaled against a baseline value; therefore the corresponding value of epsilon for Cost and Frequency of 0.01 is 3.2 for Frequency and 5.1 for Cost. The results of this numerical comparison follow next in Chapter 4.

CHAPTER 4

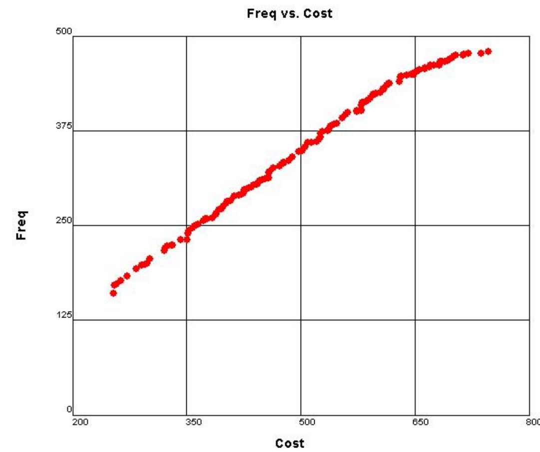
ANALYSIS AND DISCUSSION OF RESULTS

4.1 Results for Sandwich Beam Design Problem

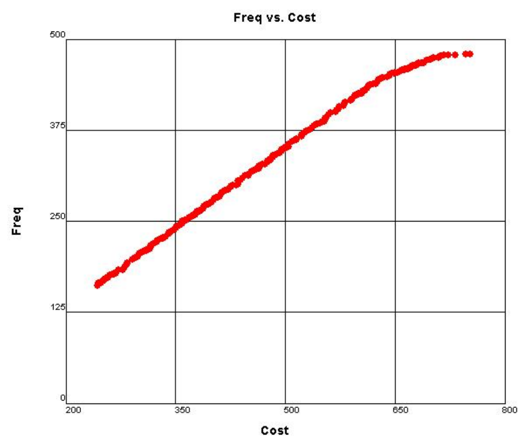
Figure 19 provides a visual comparison of the resulting Pareto frontiers from two representative trials along with the reference set for the sandwich beam design problem. Because this problem has only two objectives, the similarities in the Pareto fronts can be easily seen, particularly between Trial 10 v2 and the reference set. To show this even more clearly, Figure 20 shows all three of these fronts plotted together. While Trial 10 v2's Pareto frontier is not as complete as the reference set, it spans the entire length of the reference set and appears to be in close agreement. Despite having a relatively low ε-performance as shown by Table 10 (detailed ε-performance results of sandwich beam Trials 1-5) and Table 11 (average ε-performance results of sandwich beam Trials 1-5), Trial 3 v1 appears to be a good approximation to the reference Pareto set at least along the straight segment of the frontier. Trial 10 v2 on the other hand, does have a very high ε-performance as shown by Table 12 and Table 13, detailed and average ε-performance results of Trials 6-10, respectively.



(a) Trial 3 v1



(b) Trial 10 v2



(c) Reference

Figure 19. Example of Visual Comparisons of Resulting Pareto Frontiers from Sandwich Beam Design Problem Trials

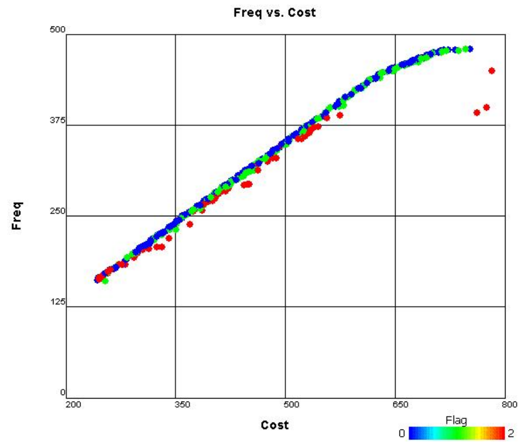


Figure 20. Sandwich Beam Design Problem Reference Pareto Front (Blue), Trial 10 v2 Pareto Front (Green), Trial 3 v1 Pareto Front (Red)

Table 10. ε-performance Results Trials 1-5 – Sandwich Beam

5,000 Point Trials									
	Trial 1			Trial 2			***Trial 3***		
# Func. Evals	v1	v2	v3	v1	v2	v3	***v1***	***v2***	v3
	% Within ε			% Within ε			% Within ε		
500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	4.225	2.817
3000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	4.225	2.817
4000	0.000	0.000	0.000	0.000	0.000	0.000	2.817	5.634	2.817
5000	0.000	1.408	0.000	0.000	0.000	0.000	***5.634***	***5.634***	2.817
	Trial 4			Trial 5					
# Func. Evals	v1	v2	v3	v1	v2	v3			
	% Within ε			% Within ε					
500	0.000	0.000	0.000	0.000	0.000	0.000			
1000	0.000	0.000	0.000	0.000	0.000	0.000			
2000	0.000	0.000	0.000	0.000	0.000	0.000			
3000	0.000	0.000	0.000	0.000	0.000	0.000			
4000	0.000	0.000	0.000	0.000	0.000	0.000			
5000	0.000	0.000	0.000	0.000	0.000	1.408			

Table 11. Average ε-performance Results Trials 1-5 – Sandwich Beam

5,000 Point Trial Averages					
# Func. Evals	Trial 1	Trial 2	***Trial 3***	Trial 4	Trial 5
	% Within ε				
500	0.000	0.000	0.000	0.000	0.000
1000	0.000	0.000	0.000	0.000	0.000
2000	0.000	0.000	2.347	0.000	0.000
3000	0.000	0.000	2.347	0.000	0.000
4000	0.000	0.000	3.756	0.000	0.000
5000	0.469	0.000	***4.695***	0.000	0.469

Table 12. ε-performance Results Trials 6-10 – Sandwich Beam

10,000 Point Trials									
	Trial 6			Trial 7			Trial 8		
# Func. Evals	v1	v2	v3	v1	v2	v3	v1	v2	v3
	% Within ε			% Within ε			% Within ε		
500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2500	0.000	1.408	4.225	0.000	0.000	0.000	0.000	2.817	1.408
5000	2.817	1.408	4.225	0.000	0.000	0.000	0.000	4.225	5.634
7500	2.817	1.408	8.451	0.000	0.000	0.000	11.268	5.634	5.634
10000	2.817	1.408	8.451	0.000	0.000	0.000	19.718	12.676	8.451
	Trial 9			***Trial 10***					
# Func. Evals	v1	v2	v3	v1	***v2***	v3			
0	% Within ε			% Within ε					
500	0.000	0.000	0.000	0.000	0.000	0.000			
1000	0.000	0.000	0.000	0.000	0.000	0.000			
2500	0.000	0.000	0.000	9.859	0.000	0.000			
5000	0.000	0.000	0.000	19.718	9.859	16.901			
7500	0.000	0.000	0.000	30.986	28.169	32.394			
10000	0.000	0.000	0.000	38.028	***40.845***	38.028			

Table 13. Average ε-performance Results Trials 6-10 – Sandwich Beam

10,000 Point Trial Averages					
# Func. Evals	Trial 6	Trial 7	Trial 8	Trial 9	***Trial 10***
	% Within ε				
500	0.000	0.000	0.000	0.000	0.000
1000	0.000	0.000	0.000	0.000	0.000
2500	1.878	0.000	1.408	0.000	3.286
5000	2.817	0.000	3.286	0.000	15.493
7500	4.225	0.000	7.512	0.000	30.516
10000	4.225	0.000	13.615	0.000	***38.967***

An interesting and at first somewhat surprising point to note, as shown by Table 10, Table 11, Table 12, and Table 13, is that Trials 2 and 4 and Trials 7 and 9 each did not find any of the reference Pareto set. While it was expected to be shown that the user-guided trials would outperform the "blindly" searching automated trials, it is unexpected that the Pareto sampler only and unguided basic sampler only trials would not find any part of the reference Pareto frontier. However, inspection of the trade space reveals the answer; the entire trade space from Trial 7 v1 with the reference Pareto Frontier overlaid is shown in Figure 21.

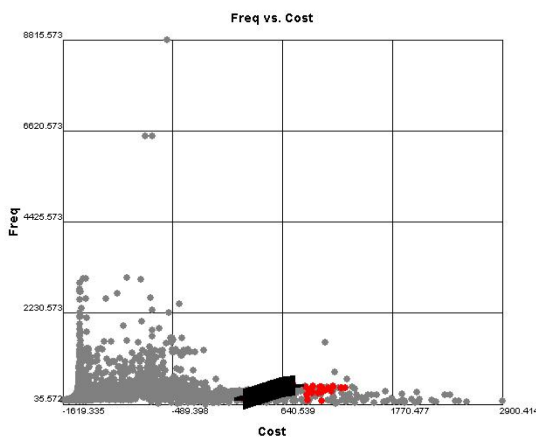
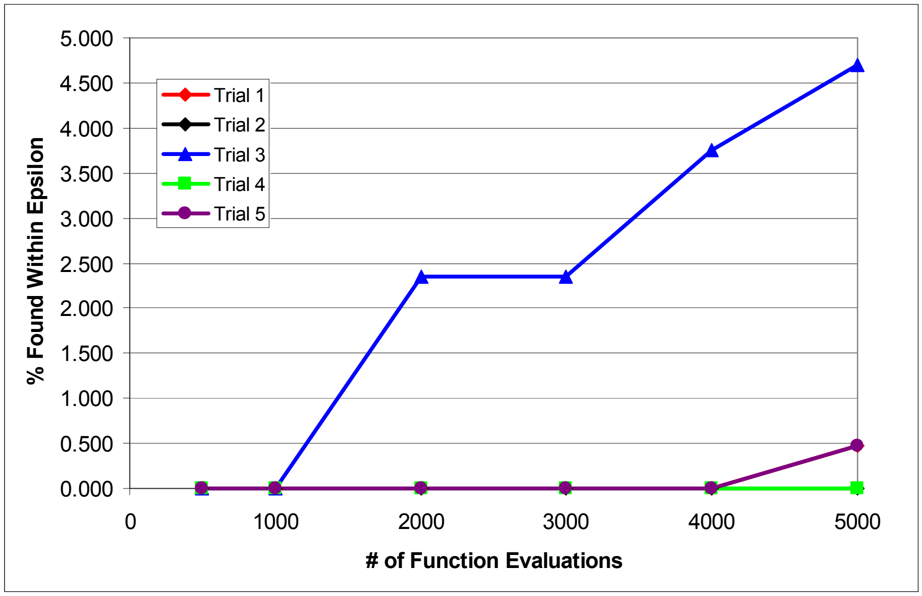


Figure 21. Trial 7 v1 Entire Trade Space Overlaid with Reference Pareto Frontier (Denoted by +), Infeasible Points (Gray), Feasible Points (Red)

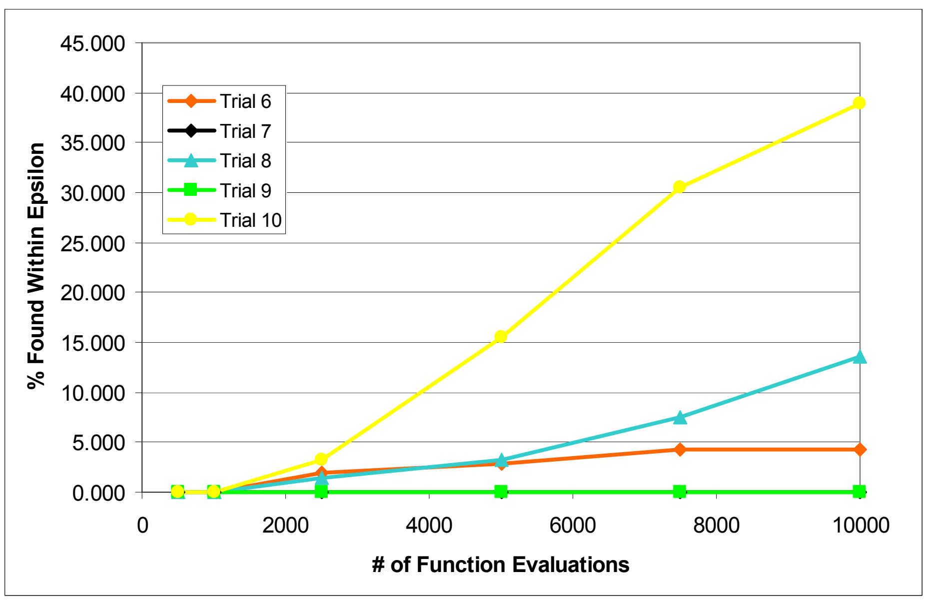
Examining Figure 21 shows how large the trade space is and how most of it contains infeasible designs. During the user-guided trials, the user is able to steer the exploration into the relatively small feasible region in the trade space, however, the "blind" trials have no way of knowing that they are sampling nothing but infeasible points. The Pareto sampler in this case would be sampling the far left edge of the trade space in an infeasible region with no ability to break free. The unguided basic sampler would rely solely on chance to find points along the Pareto frontier, which is highly unlikely given the size of the trade space. For a sense of scale, the reference Pareto frontier shown in Figure 19 is represented in Figure 21 by the black "+" symbols which are so tightly packed together given the scale that they are difficult to discern.

Few of the user trials did well, however. Trial 10 is by far the best performer of the user-guided trials, and the other 10,000 function evaluation trials did fairly well, but only Trial 3 of the 5,000 function evaluation trials shows any promise according to Table 10 and Table 12. This shows that not just any combination of steering commands will work; they need to be intelligently used so that they are effective and do not rely too heavily upon chance to generate points of use.

It is difficult to make generalizations based on this problem due to the wide variety of results, but given such a constrained problem, Table 10, Table 12, and Figure 22 show that a user-guided trial can significantly outperform a "blind" search in which no constraint handling method is applied. The user can make changes on the fly as the exploration process unfolds, whereas the automated search, in this case, cannot. Figure 22 gives a way of visualizing the results contained in Table 11 and in Table 13 and gives a sense of how quickly the user-guided trials are converging to the reference Pareto frontier, while showing that the Pareto sampler only Trials 2 and 7 and unguided basic sampler only Trials 4 and 9 found none of the reference set design points over the full range of allotted function evaluations.



(a) Results from Trials 1-5



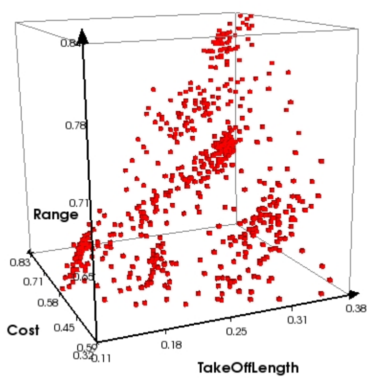
(b) Results from Trials 6-10

Figure 22. Evolution of Pareto Frontiers for the Sandwich Beam Design Problem, Trial Average

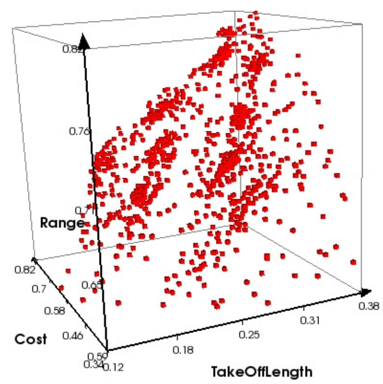
4.2 Results for Aircraft Wing Design Problem

While a visual comparison among Pareto frontiers is not as easily done as with the sandwich beam test problem, Figure 23 provides a visual comparison of the resulting

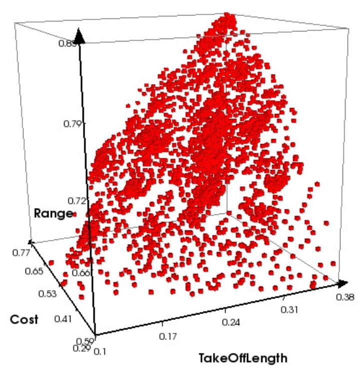
Pareto frontiers for the aircraft wing design problem for representative trials along with the reference set. The similar overall shape of the Pareto frontiers is readily apparent, and it appears that Trial 6 v2 is closer to the reference set than Trial 5 v1. This is supported by the results shown in Table 14 and Table 15, detailed ε-performance results of Trials 1-5 and Trials 6-10, respectively.



(a) Trial 5 v1



(b) Trial 6 v2



(c) Reference

Figure 23. Example of Visual Comparisons of Resulting Pareto Frontiers

Whereas the sandwich beam problem only had a few standout trials, every wing trial, although not every version, found part of the reference Pareto frontier. Both of the unguided trials in each set had at least some success. For the 5,000 function evaluation trials Table 16 shows the user-guided trials (Trials 1, 3, and 5) found, on average, 1.72%

- 4.23% of the reference set, the unguided trials (Trials 2 and 4) found, on average, 0.13% - 1.46%. The 10,000 function evaluation trials, as expected, performed even better. Table 17 shows that the user-guided trials found 3.57% - 11.77% of the reference Pareto set whereas the unguided trials found 0.53% - 3.57%. This indicates that, like the previous problem showed, not just any combination of visual steering commands will perform well, but if done so intelligently, they could produce up to a 22x - 32x increase in the number of Pareto points found over random searching and up to a 3x increase in the number of Pareto points found compared to an automated sampler.

Table 14. ε-performance Results Trials 1-5 – Aircraft Wing

5,000 Point Trials									
	Trial 1			Trial 2			Trial 3		
# Func. Evals	v1	v2	v3	v1	v2	v3	v1	v2	v3
	% Within ε			% Within ε			% Within ε		
500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1000	0.000	0.000	0.000	0.000	0.000	0.000	0.397	0.000	0.000
2000	1.587	0.000	0.000	0.794	0.000	0.000	0.794	0.000	0.794
3000	1.587	0.000	0.000	1.587	0.397	0.397	2.381	0.000	1.190
4000	1.587	0.000	0.397	2.381	0.397	0.397	2.381	0.000	1.984
5000	3.571	0.397	2.381	2.778	1.190	0.397	2.778	0.000	2.381
	Trial 4			***Trial 5***					
# Func. Evals	v1	v2	v3	***v1***	v2	v3			
	% Within ε			% Within ε					
500	0.000	0.000	0.000	0.000	0.000	0.000			
1000	0.000	0.000	0.000	0.000	0.000	0.000			
2000	0.000	0.000	0.000	1.587	0.000	0.794			
3000	0.000	0.000	0.000	2.778	0.000	1.984			
4000	0.000	0.000	0.397	3.968	1.587	1.984			
5000	0.000	0.000	0.397	***6.349***	2.381	3.968			

Table 15. ε-performance Results Trials 6-10 – Aircraft Wing

10,000 Point Trials									
	Trial 6			Trial 7			Trial 8		
# Func. Evals	v1	***v2***	v3	v1	v2	v3	v1	v2	v3
	% Within ε			% Within ε			% Within ε		
500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1000	0.397	1.984	0.000	0.000	0.000	0.000	0.794	0.000	0.397
2500	1.190	2.381	1.190	0.397	0.000	0.000	5.159	0.397	1.190
5000	6.349	4.365	6.746	0.397	0.794	3.968	6.746	0.397	3.175
7500	11.508	9.127	8.333	0.794	1.587	6.349	7.143	5.556	6.349
10000	11.905	***12.302***	11.111	1.587	1.587	7.540	13.492	8.333	10.318
	Trial 9			Trial 10					
# Func. Evals	v1	v2	v3	v1	v2	v3			
0	% Within ε			% Within ε					
500	0.000	0.000	0.000	0.000	0.000	0.000			
1000	0.000	0.000	0.000	0.000	0.397	0.000			
2500	0.000	0.000	0.000	0.397	0.794	0.000			
5000	0.000	0.397	0.000	1.190	1.190	1.190			
7500	0.000	1.190	0.397	3.571	2.778	1.190			
10000	0.000	1.190	0.397	3.968	4.365	2.381			

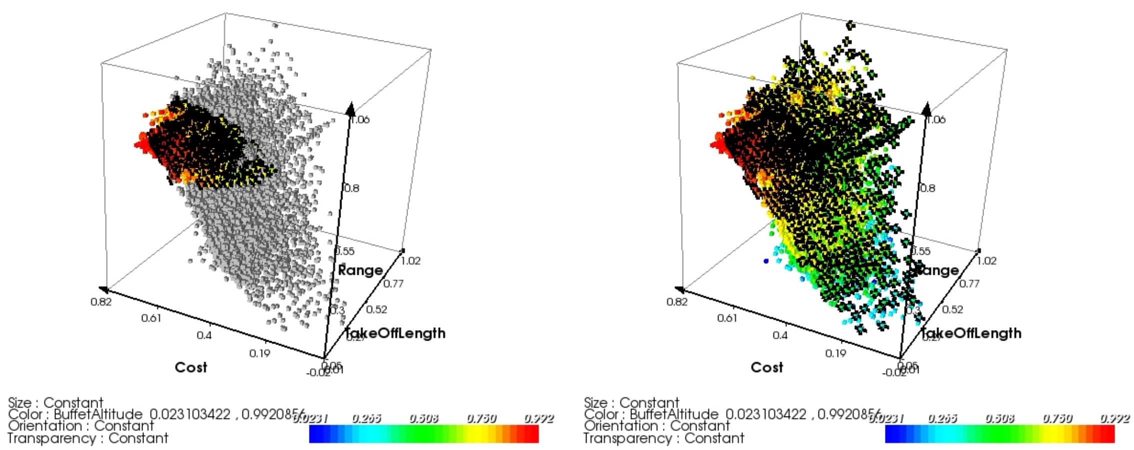
Table 16. Average ε-performance Results Trials 1-5 – Aircraft Wing

5,000 Point Trial Averages					
# Func. Evals	Trial 1	Trial 2	Trial 3	Trial 4	***Trial 5***
	% Within ε				
500	0.000	0.000	0.000	0.000	0.000
1000	0.000	0.000	0.132	0.000	0.000
2000	0.529	0.265	0.529	0.000	0.794
3000	0.529	0.794	1.190	0.000	1.587
4000	0.661	1.058	1.455	0.132	2.513
5000	2.116	1.455	1.720	0.132	***4.233***

Table 17. Average ε-performance Results Trials 6-10 – Aircraft Wing

10,000 Point Trial Averages					
# Func. Evals	***Trial 6***	Trial 7	Trial 8	Trial 9	Trial 10
	% Within ε				
500	0.000	0.000	0.000	0.000	0.000
1000	0.794	0.000	0.397	0.000	0.132
2500	1.587	0.132	2.249	0.000	0.397
5000	5.820	1.720	3.439	0.132	1.190
7500	9.656	2.910	6.349	0.529	2.513
10000	***11.773***	3.571	10.714	0.529	3.571

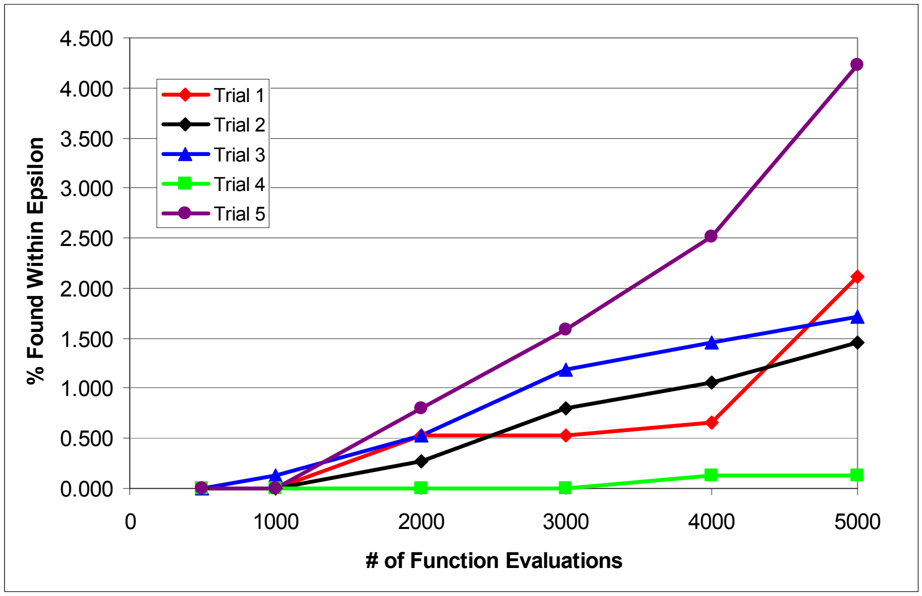
Figure 25 gives a visual representation of the results in Table 16 and Table 17. It shows that the increase is generally present during the full course of the allotted number of function evaluations, with Trial 10 performing nearly identical to Trial 7, and Trial 1 alternating with Trial 2. The trials that performed considerably better than the automated trials did so during the whole course of the trials. It is also very clear that all user-guided trials outperformed the unguided basic sampler trials. It should also be pointed out that the automated trials suffer from a similar problem to that seen in the sandwich beam problem trials, namely, a large portion of the trade space is actually infeasible as seen in Figure 24.



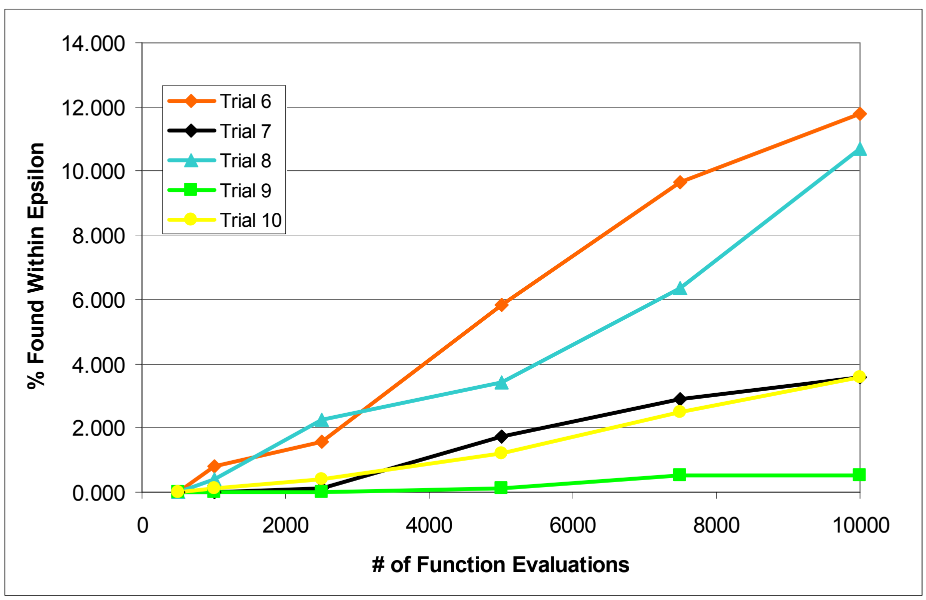
(a) Feasible (Colored) Pareto Front, Infeasible points (Gray) (b) Combined Pareto Front

Figure 24. Feasible vs. Infeasible Trade Space (Pareto Points Denoted by +)

The Pareto sampler would be attempting to advance the entire frontier (Figure 24b) while the desired frontier is only a small portion (Figure 24a). This again, shows the strength of the user-guided trials. The user can adapt the steering commands to focus in on only the feasible region of interest.



(a) Results from Trials 1-5

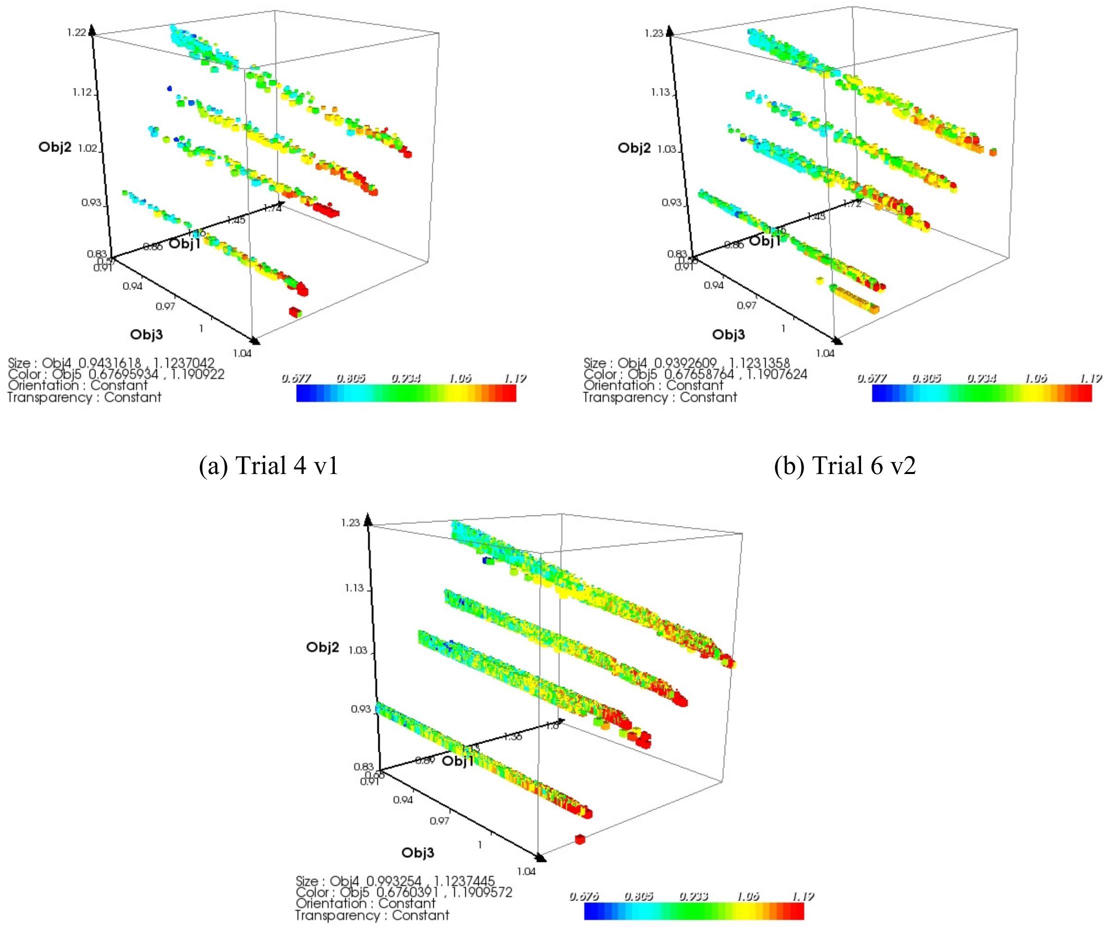


(b) Results from Trials 6-10

Figure 25. Evolution of Pareto Frontiers for the Aircraft Wing Design Problem, Trial Average

4.3 Results for Vehicle Configuration Design Problem

Figure 26 provides a visual comparison of the resulting Pareto frontiers of representative trials along with the reference set for the VCM problem. While it is difficult to make comparisons in 5-D, it can seen that all three plots share similar characteristics.



(a) Trial 4 v1

(b) Trial 6 v2

(c) Reference

Figure 26. Example of Visual Comparisons of Resulting VCM Pareto Frontiers

Table 18 summarizes the detailed results of each trial using the ε-performance metric for the 5,000 function evaluation trials and the MOGA (up to 5,000 function evaluations) while Table 19 lists the averages of the three versions. As discussed in Chapter 3.4, the reported results are obtained by comparing each trial's resulting Pareto front from each version to the reference set obtained from combining all fronts, which is the best approximation of the "true" Pareto frontier that can be obtained. Not surprisingly, the percentage of designs found within epsilon of the reference solutions is very low for every trial. This is likely a result of the objective space being 5-D, which makes it very difficult to find two designs that fall within 0.01 of each other in all five objectives. However, as stated before, if the epsilon value is increased, it makes comparisons meaningless because every set is substantially reduced; an example is shown in Table 20. It can be seen that when using only 5,000 points, the user trials are able to obtain on average 0.41% - 1.70% of the reference set whereas the automated, "blindly" searching Trial 5 and MOGA, only found 0.26% - 0.36% of the reference set, on average. From Table 19 it can be seen that the user trials were able to find up to nearly 5x as many Pareto points as the MOGA (1.70% compared to 0.36%) and 7x as many as a random search.

Table 18. ε-performance Results Trials 1-5 and MOGA – VCM

5,000 Point Trials									
	Trial 1			***Trial 2***			Trial 3		
# Func. Evals	v1	v2	v3	v1	***v2***	v3	v1	v2	v3
	% Within ε			% Within ε			% Within ε		
500	0.000	0.000	0.155	0.155	0.000	0.155	0.000	0.155	0.000
1000	0.000	0.155	0.309	0.155	0.000	0.464	0.000	0.309	0.000
2000	0.309	0.309	0.464	0.155	0.464	0.618	0.000	0.618	0.155
3000	1.391	0.464	0.618	0.155	0.927	0.773	0.618	0.773	0.309
4000	1.546	0.464	0.618	0.464	2.782	0.773	1.391	1.700	0.773
5000	1.546	0.464	1.391	0.618	***2.937***	1.546	1.391	2.164	0.773
	Trial 4			Trial 5			MOGA		
# Func. Evals	v1	v2	v3	v1	v2	v3	v1	v2	v3
	% Within ε			% Within ε			% Within ε		
500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1000	0.000	0.000	0.155	0.000	0.000	0.000	0.000	0.000	0.000
2000	0.464	0.309	0.155	0.000	0.155	0.000	0.155	0.000	0.000
3000	0.464	0.309	0.155	0.000	0.464	0.000	0.155	0.000	0.155
4000	0.618	0.464	0.155	0.155	0.464	0.000	0.618	0.000	0.309
5000	0.618	0.464	0.155	0.155	0.618	0.000	0.773	0.000	0.309

Table 19. Average ε-performance Results Trials 1-5 and MOGA – VCM

5,000 Point Trial Averages						
# Func. Evals	Trial 1	***Trial 2***	Trial 3	Trial 4	Trial 5	MOGA
	% Within ε					
500	0.052	0.103	0.052	0.000	0.000	0.000
1000	0.155	0.206	0.103	0.052	0.000	0.000
2000	0.361	0.412	0.258	0.309	0.052	0.052
3000	0.824	0.618	0.567	0.309	0.155	0.103
4000	0.876	1.340	1.288	0.412	0.206	0.309
5000	1.133	***1.700***	1.443	0.412	0.258	0.361

Table 20. Example of ε and ε-performance Values

Trial 8 v1		
ε	ε-performance	# Pareto Points in Reference Set
0.001	0.13	10,162
0.005	0.67	2,071
0.01	4.17	647
0.05	26.70	30
0.1	50.00	10
0.2	66.70	3
0.3	100.00	2

As expected, Table 21 (detailed 10,000 function evaluation trial results) and Table 22 (average 10,000 function evaluation trial results) show that the 10,000 function evaluation trials perform even better than the 5,000 function evaluation trials in nearly every case, with Trial 8 performing the best. When given 10,000 function evaluations, the user trials find on average 1.24% - 3.50% of the reference set, while the trials without a human-in-the-loop find 0.36% - 0.57% on average, meaning the user trials are able to find up to 6x as many Pareto points as the MOGA and 10x as many as a random search.

Table 21. ε-performance Results Trials 6-10 and MOGA – VCM

10,000 Point Trials									
	Trial 6			Trial 7			***Trial 8***		
# Func. Evals	v1	v2	v3	v1	v2	v3	***v1***	v2	v3
	% Within ε			% Within ε			% Within ε		
500	0.000	0.000	0.155	0.464	0.000	0.000	0.000	0.155	0.000
1000	0.000	0.309	0.155	1.700	0.000	0.000	0.000	0.309	0.155
2500	0.155	0.927	0.927	1.700	0.155	0.000	0.464	1.236	0.464
5000	0.773	0.927	2.782	2.318	0.309	0.464	2.628	2.937	1.236
7500	1.700	1.236	2.937	2.473	0.927	1.236	3.709	2.937	1.700
10000	2.009	1.236	2.937	3.555	2.009	1.700	***4.173***	3.709	2.628
	Trial 9			Trial 10			MOGA		
# Func. Evals	v1	v2	v3	v1	v2	v3	v1	v2	v3
0	% Within ε			% Within ε			% Within ε		
500	0.155	0.309	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1000	0.155	0.464	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2500	0.309	0.927	0.155	0.000	0.000	0.000	0.155	0.000	0.155
5000	0.464	1.082	0.773	0.000	0.155	0.155	0.773	0.000	0.309
7500	0.464	1.236	1.236	0.000	0.309	0.464	0.773	0.155	0.618
10000	0.773	1.546	1.391	0.000	0.464	0.618	0.927	0.155	0.618

Table 22. Average ε-performance Results Trials 6-10 and MOGA – VCM

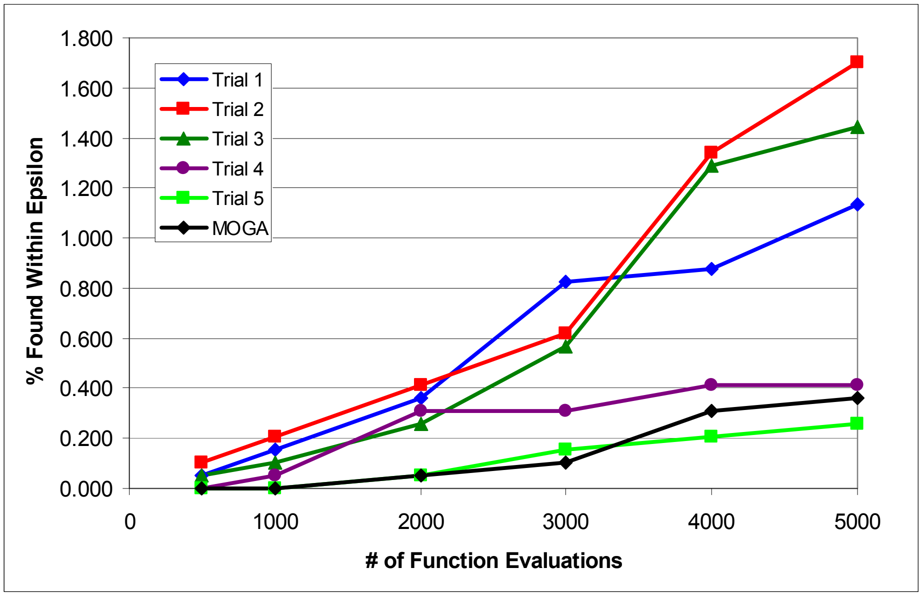
10,000 Point Trial Averages						
# Func. Evals	Trial 6	Trial 7	***Trial 8***	Trial 9	Trial 10	MOGA
	% Within ε					
500	0.052	0.155	0.052	0.155	0.000	0.000
1000	0.155	0.567	0.155	0.206	0.000	0.000
2500	0.670	0.618	0.721	0.464	0.000	0.103
5000	1.494	1.030	2.267	0.773	0.103	0.361
7500	1.958	1.546	2.782	0.979	0.258	0.515
10000	2.061	2.421	***3.503***	1.236	0.361	0.567

To gain more insight into the performance of each trial as well as the evolution of solutions toward the Pareto frontier, plots of the ε-performance metric at a series of intervals leading up to the allocated number of function evaluations are made. In particular, Figure 27a shows the average performance of Trials 1-5 at 500, 1000, 2000, 3000, 4000, and 5000 function evaluations; Figure 27b shows a similar progression for

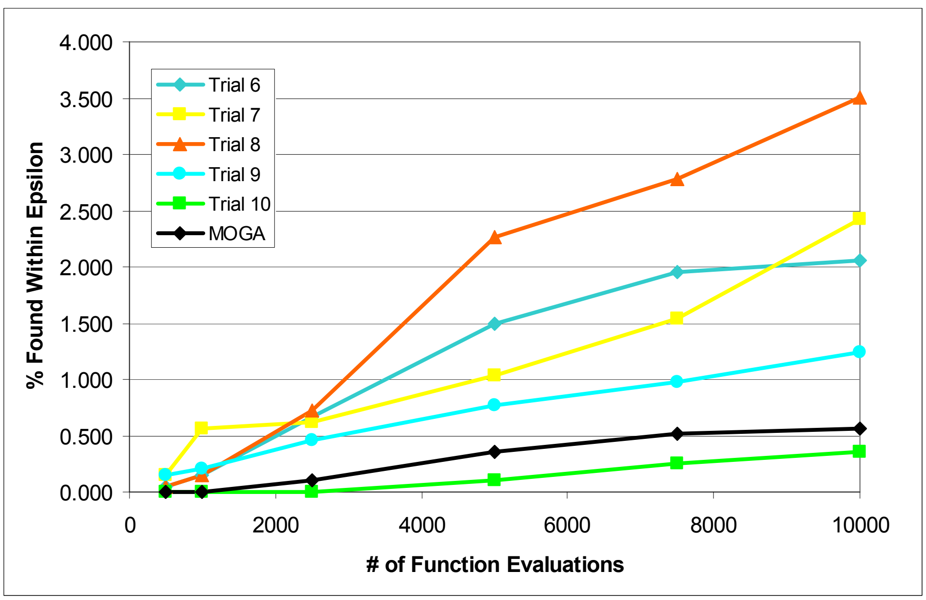
Trials 6-10 at 500, 1000, 2500, 5000, 7500, and 10,000 function evaluations. In both figures, solutions from the exhaustive MOGA are also plotted based on its convergence history; so, for example, the ε-performance metric value plotted at the 500 function evaluation point indicates how well the MOGA has found the Pareto frontier by the time it has executed 500 of its 80,000 function evaluations.

While the results from Table 19 and Table 22 may not have been too convincing, Figure 27 clearly illustrates the benefit of having the user "in-the-loop" during the optimization process. On average (with the exception of the basic sampler only trials, Trials 5 and 10), all trials out-perform the exhaustive MOGA in terms of the percentage of solutions found on the Pareto frontier (i.e., the reference set) for a given number of function evaluations. In Figure 27a, the MOGA has obtained fewer than 0.40% of the Pareto frontier in its first 5000 function evaluations compared to 0.41% - 1.70% in Trials 1-5. Likewise, even when the number of function evaluations has doubled to 10,000, the MOGA has still found less than 0.60% of the Pareto frontier solutions compared to the 1.24% - 3.50% obtained in Trials 6-10 as indicated in Figure 27b. In both cases, this represents a 5x - 6x increase in the number of Pareto solutions that are obtained when the human is allowed to visualize and "steer" the optimization process. This increase is obtained consistently, regardless of the particular combination of visual steering commands that are used or the designer who is implementing them. While the percentage of the Pareto frontier found is fairly low (and lower than results from the other two test problems) for even the best performing user trials, it should still be considered significant by realizing that the user trials were only allotted ~5,000 to ~10,000 function evaluations whereas the reference set is the culmination of approximately 465,000 function evaluations (MOGA 80,000 x 3 runs + 5,000 x 5 trials x 3 versions + 10,000 x 5 trials x 3 versions).

There are a few important implications that can be taken from the results of the three test problems. In particular, the results show that user-guided trade space exploration with visual steering commands is capable of outperforming automated, "blindly" searching samplers. The results are even more dramatic when the automated samplers had difficultly finding feasible points, but it should be noted that not all user-guided trials did well – simply employing steering commands will not necessarily lead to finding Pareto or feasible designs. The commands were best used in conjunction with observations made about the trade space and relationships among design variables to guide the search in the desired direction. In general, the trials that performed the best were those that made effective use of the visual steering commands to sample in regions of feasibility, particularly, by placing attractors or adjusting the bounds of design variables when using the basic sampler. Chapter 5 summarizes the results and limitations of this research and outlines suggestions for future work.



(a) Results from Trials 1-5 and Exhaustive MOGA Search



(b) Results from Trials 6-10 and Exhaustive MOGA Search

Figure 27. Evolution of Pareto Frontiers for VCM, Trial Average and the Exhaustive MOGA Search

CHAPTER 5

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

As stated earlier, trade space exploration is a promising alternative decision-making paradigm that provides a visual and more intuitive means for formulating, adjusting, and ultimately solving design optimization problems. The results of this study indicate that the visual steering commands – depending on the complexity of the test problem – can provide a 3x - 6x increase in the number of Pareto solutions that are obtained when the human is "in-the-loop" during the optimization process compared to an automated sampler. The improvements are even more dramatic in cases where automated samplers have a difficult time finding feasible solutions. In addition, user-guided trials can provide a 10x - 32x speed up in the number of Pareto solutions obtained over random searching. As such, this study provides empirical evidence of the benefits that interactive visualization-based strategies can provide in support of engineering design optimization and decision-making.

One limitation of note for the study performed in this thesis is the virtually unlimited number of combinations of visual steering commands that could be utilized on these test problems. The results show that the user-guided trials were more effective than the automated samplers; however, their performance may have been hindered by the specific combination of commands used – experienced users may identify better combinations of the visual steering commands as they explore the trade space while less experienced users may not yet know how to use visual steering commands as effectively. Different ways of using the commands may have resulted in even better performing trials, and additional trials are needed on a wider variety of test problems by users with varying levels of expertise. Another possible limitation is the number of points allocated for the trials. Continuing the trials for more evaluations could help identify at what point the results start to plateau in terms of performance, providing useful information for stopping the exploration process.

There are several possible extensions of this work. Additional metrics should be considered for comparing the solutions in the resulting Pareto frontiers in terms of both the design variables (inputs) as well as the objective function values (outputs). A multi-metric strategy would be useful in not only assessing the completeness of the Pareto frontiers more thoroughly but also providing guidance to the users if computed in real-time during the trade space exploration process. Constraint handling methods should be added to the automated samplers to see if their performance improves in highly constrained test problems. An extension of that would be to examine, in the event the automated sampler's performance was able to meet that of the user-guided trials when implementing a constraint handling strategy, how quick and effective the user-guided visual steering commands would be at searching various regions of the trade space by utilizing different commands, as compared to the automated samplers since they would need the constraints redefined. Preliminary investigations of constraint handling strategies have been shown to be very effective [34]. Finally, the study should also be repeated with even more test problems of different sizes and complexity as well as with users with different experience levels to demonstrate how widely applicable – and beneficial – the trade space exploration process is.

REFERENCES

[1] Balling, R., 1999, "Design by Shopping: A New Paradigm?" *Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization (WCSMO-3)*, Buffalo, New York, University at Buffalo, 295-297.

[2] Wilson, T. D. and Schooler, J.W., 1991, "Thinking Too Much: Introspection can Reduce the Quality of Preferences and Decisions," *Journal of Personality and Social Psychology*, 60(2), 181-192.

[3] Shanteau, J., 1992, "Competence in Experts: The Role of Task Characteristics," *Organizational Behavior and Human Decisions*, 53(2), 252-266.

[4] Stump, G., Lego, S., Yukish, M., Simpson, T. W. and Donndelinger, J. A., 2007, "Visual Steering Commands for Trade Space Exploration: User-Guided Sampling with Example", *ASME Design Engineering Technical Conferences - Design Automation Conference*, Las Vegas, NV, ASME, DETC2007/DAC-34684.

[5] Stump, G., Yukish, M. and Simpson, T. W., 2004, "The ARL Trade Space Visualizer: An Engineering Decision-Making Tool", *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, AIAA, AIAA-2004-4568.

[6] Stump, G., Yukish, M., Simpson, T. W., Harris, E. N. and O'Hara, J. J., 2004, "Trade Space Exploration of Satellite Datasets Using a Design by Shopping Paradigm", *IEEE Aerospace Conference*, Big Sky, MT, IEEE, IEEE-1039-04.

[7] Buja, A., McDonald, J. A., Michalak, J. and Stuetzle, W., 1991, "Interactive Data Visualization Using Focusing and Linking", *Proceedings of the IEEE Conference on Visualization '91*, San Diego, CA, IEEE Computer Society Press, 156-163.

[8] Becker, R. A. and Cleveland, W. S., 1987, "Brushing Scatterplots," *Technometrics*, 29(1), 127-142.

[9] Wang, G. G. and Shan, S., 2007, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," *ASME Journal of Mechanical Design*, 129(4), 370-380.

[10] Wright, H., Brodlie, K. and David, T., 2000, "Navigating High-Dimensional Spaces to Support Design Steering", *Proceedings of IEEE Visualization 2000*, Salt Lake City, UT, IEEE Computer Society Press, 291-296.

[11] Kesavadas, T. and Sudhir, A., 2000, "Computational Steering in Simulation of Manufacturing Systems", *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, IEEE, 2654-2658.

[12] Messac, A. and Chen, X., 2000, "Visualizing the Optimization Process in Real-Time Using Physical Programming," *Engineering Optimization*, 32(6), 721-747.

[13] Winer, E. H. and Bloebaum, C. L., 2001, "Visual Design Steering for Optimization Solution Improvement," *Structural Optimization*, 22(3), 219-229.

[14] Winer, E. H. and Bloebaum, C. L., 2002, "Development of Visual Design Steering as an Aid in Large-Scale Multidisciplinary Design Optimization. Part I: Method Development," *Structural and Multidisciplinary Optimization*, 23(6), 412-424.

[15] Scott, S. D., Lesh, N. and Klau, G. W., 2001, Investigating Human-Computer Optimization, CHI. Minneapolis, MN.

[16] Miettinen, K. and Makela, M., 2006, "Synchronous approach in interactive multiobjective optimization," *European Journal of Operational Research*, 170, 909-922.
[17] Madar, J., Abonyi, J. and Szeifert, F., 2005, "Interactive Particle Swarm Optimization", *International Conference on Intelligent Systems Design and Applications*, IEEE.
[18] Michalek, J. and Papalambros, P., 2002, "Interactive Design Optimization of Architectural Layouts," *Eng. Opt.*, 34(5), 485-501.
[19] Price, K., Storn, R. and Lampinen, J., 2005, *Differential Evolution - A Practical Approach to Global Optimization*, Berlin, Springer.
[20] Robic, T. and Filipi, B., 2005, "DEMO: Differential Evolution for Multiobjective Optimization." *Proceedings of Conference on Evolutionary Multiobjective Optimization*.
[21] Messac, A., 1996, "Physical Programming: Effective Optimization for Computational Design," *AIAA Journal*, 34(1), 149-158.
[22] Simpson, T. W. and Meckesheimer, M., 2004, "Evaluation of a Graphical Design Interface for Design Space Visualization", *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Palm Springs, CA, AIAA.
[23] Donndelinger, J., Ferguson, S. and Lewis, K., 2006, "Exploring Mass Trade-Offs in Preliminary Vehicle Design Using Pareto Sets", *11th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Portsmouth, VA, AIAA, AIAA-2006-7056.
[24] Ferguson, S., Gurnani, A., Donndelinger, J. and Lewis, K., 2005, "A Study of Convergence and Mapping in Multiobjective Optimization Problems", *ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Long Beach, CA, ASME, Paper No. DETC2005/CIE-84852.
[25] Ferguson, S., Gurnani, A., Donndelinger, J. and Lewis, K., 2005, "An Approach to Feasibility Assessment In Preliminary Design", *ASME Design Engineering Technical Conferences - Design Automation Conference*, Long Beach, CA, ASME, Paper No. DETC2005/CIE-84853.
[26] Sanchis, J., Martinez, M. and Blasco, X., 2008, "Multi-objective engineering design using preferences," *Engineering Optimization*, 40(3), 253-269.
[27] Dym, C. L., Wood, W. H. and Scott, M. J., 2006, "On the Legitimacy of Pairwise Comparisons", *Decision Making in Engineering Design* ASME, 135-143.
[28] Wu, J. and Azarm, S., 2001, "Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set," *ASME Journal of Mechanical Design*, 123(1), 18-25.
[29] Okabe, T., Jin, Y. and Sendhoff, B., 2003, "A Critical Survey of Performance Indices for Multi-Objective Optimisation," *Evolutionary Computation*, 2, 878-885.
[30] Zitzler, E., 1999, Evolutionary Algorithms for Multiobjective: Methods and Applications. Zuerich, Switzerland, Swiss Federal Institute of Technology.

[31] Tang, Y., Reed, P.M. and Wagener, T., 2005, "How Effective and Efficient are Mulitobjective Evolutionary Algorithms at Hydrologic Model Calibration," *Hydrology and Earth System Sciences Discussions*, 2, 2465-2520.

[32] Kollat, J. B. and Reed, P. M., 2005, "Comparing State-of-the-Art Evolutionary Multi-Objective Algorithms for Long-Term Groundwater Monitoring Design," *Advances in Water Resources*, 29(6), 792-807.

[33] Kollat, J. B. and Reed, P. M., 2005, "The Value of Online Adaptive Search: A Performance Comparison of NSGAII, e-NSGAII, and eMOEA", *Lecture Notes in Computer Science*, Springer Berlin, 3410.

[34] Simpson, T. W., Carlsen, D. E., Congdon, C. D., Stump, G. and Yukish, M. A., 2008, "Trade Space Exploration of a Wing Design Problem Using Visual Steering and Multi-Dimensional Data Visualization", *4th AIAA Multidisciplinary Design Optimization Specialist Conference*, Schaumburg, IL, AIAA, 2008, AIAA-2008-2139.