

DETC2008/DAC-49681

EVALUATING THE PERFORMANCE OF VISUAL STEERING COMMANDS FOR USER-GUIDED PARETO FRONTIER SAMPLING DURING TRADE SPACE EXPLORATION

Dan Carlsen¹, Matthew Malone², Josh Kollat³, and Timothy W. Simpson^{4*}
The Pennsylvania State University
University Park, PA 16802 USA

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, designs. In this paper, we compare the performance of different combinations of visual steering commands implemented by two users to a multi-objective genetic algorithm that is executed “blindly” on the same problem with no human intervention. The results indicate that the visual steering commands – regardless of the combination in which they are invoked – provide a 4x -7x increase in the number of Pareto solutions that are obtained when the human is “in-the-loop” during the optimization process. As such, this study provides the first empirical evidence of the benefits of interactive visualization-based strategies to support engineering design optimization and decision-making. Future work is also discussed.

Keywords: Visualization, Pareto frontier, multi-objective optimization, genetic algorithm.

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

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, like consumers, want to “shop” to gain intuition about trades, what is feasible and what is not, and to learn about their alternatives first before making 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]. Similar findings have been reported in other fields. For instance, Wilson and Schooler [2] have shown that people do worse at some decision tasks when asked to analyze the reasons for their preferences or evaluate all 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.

This paper presents results from ongoing research that is formalizing methods, tools, and procedures to support trade space exploration. In particular, in this paper we empirically assess the performance of visual steering commands – visually-specified controls that allow designers to “steer” an optimization algorithm – that were introduced in a previous paper [4]. This is achieved by comparing the performance of two users employing different combinations of the visual steering commands to a multi-objective genetic algorithm that is executed “blindly” on the same problem with no human intervention. Related research in computational steering is

¹ Graduate Research Assistant, Mechanical & Nuclear Engineering

² Undergraduate Research Assistant, Mechanical & Nuclear Engineering

³ Graduate Research Assistant, Civil and Environmental Engineering

^{4*} Professor of Mechanical and Industrial Engineering and Engineering Design. Member ASME. **Corresponding Author.** 314D Leonhard Building, Penn State University, University Park, PA 16802 USA. Phone/fax: (814)863-7136/4745. Email: tw8@psu.edu

discussed next before reviewing the visual steering commands available in our multi-dimensional data visualization software in Section 3. Section 4 describes the test problem used in this work and the experimental set-up for our study. The results and findings are discussed in Section 5, and future work is outlined in Section 6.

2 REVIEW OF RELATED WORK

In the visualization community, interactive optimization-based methods fall primarily into the area of computational steering whereby users (e.g., designers) interact with a model or simulation during the optimization process to help “steer” the search process toward what looks like an optimal solution. The designer observes some sort of a visual representation of the optimization process and then uses intuition, heuristics, and/or some other methods to adjust the search to move toward a design that may not have been intuitive at the beginning of the process. For instance, Wright, et al. [5] applied computational steering to design the geometry and select the grade of glass for a furnace. Kesavadas and Sudhir [6] created large-scale manufacturing simulations by allowing users to make quick changes “on-the-fly” and continue with the simulation. Messac and Chen [7] proposed an interactive visualization method wherein the progress of the optimization is visualized – but not steered – throughout the process. Finally, Visual Design Steering [8,9] 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.

Scott, et al. [10] recently proposed that including humans “in the loop” throughout the decision-making process improves the outcome. 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 neither 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 [11]. 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 [10].

Madar, et al. [12] 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 [13] 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 because of its subjective nature, but the usefulness of it in complex system design conceptualization requires further exploration.

3 OVERVIEW OF VISUALIZATION SOFTWARE AND VISUAL STEERING COMMANDS

To support trade space exploration, researchers at the Applied Research Laboratory (ARL) and Penn State have developed the ARL Trade Space Visualizer (ATSV) [14,15], 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 [16], and brushing [17]. Figure 1 shows several examples of its data visualization capability. The glyph plot (left) can display up to seven dimensions by assigning variables to the x-axis, y-axis, z-axis, position, 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 the samples in each dimension.

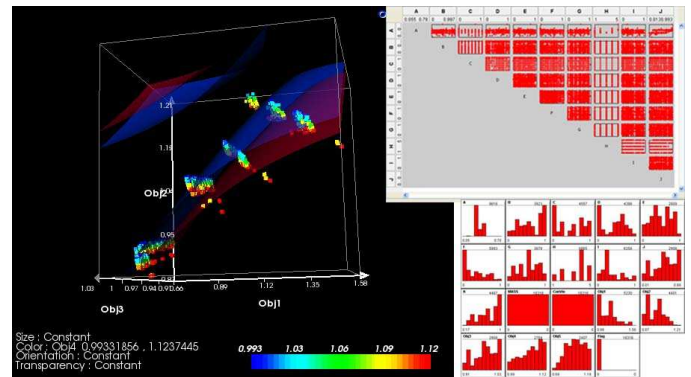
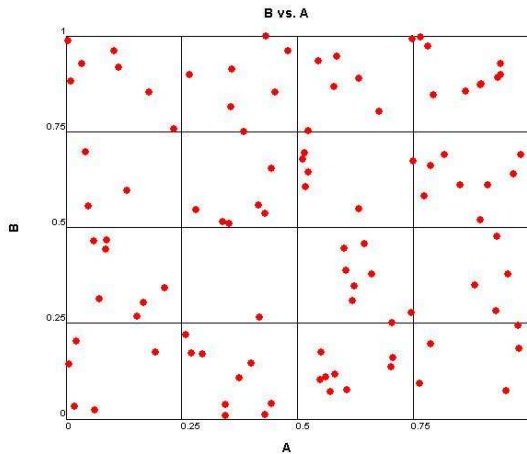


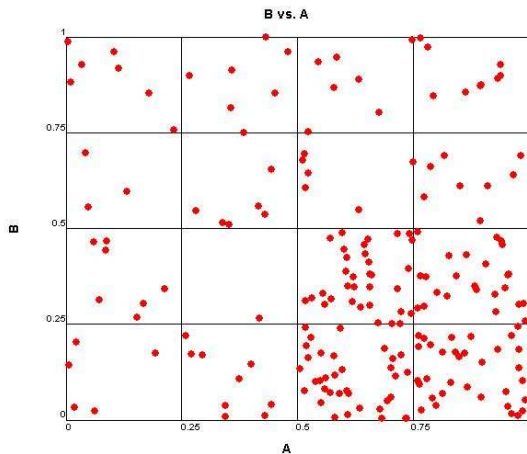
Figure 1. Three Displays of Data in ATSV

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 [18]. Once this link is in place, ATSV provides a suite of controls to help designers navigate and explore the trade space, including visual steering commands to (1) randomly sample the design space, (2) search near a point of interest, (3) search in a direction of preference, or (4) search for the Pareto frontier [4]. A brief summary of each follows.

1) *Design space samplers* are used to populate the trade space and are typically invoked if there is no initial data available. 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

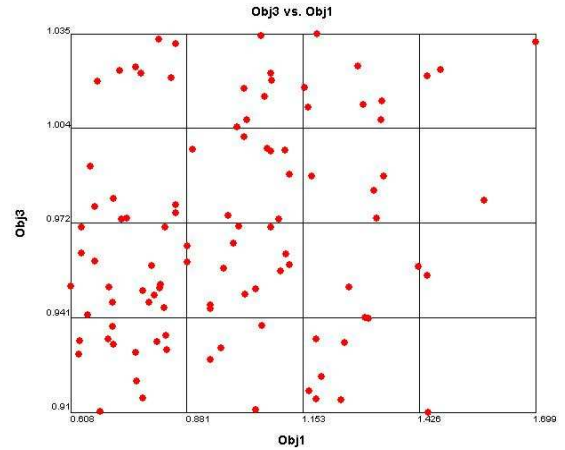


(b) 100 new samples in reduced region of interest

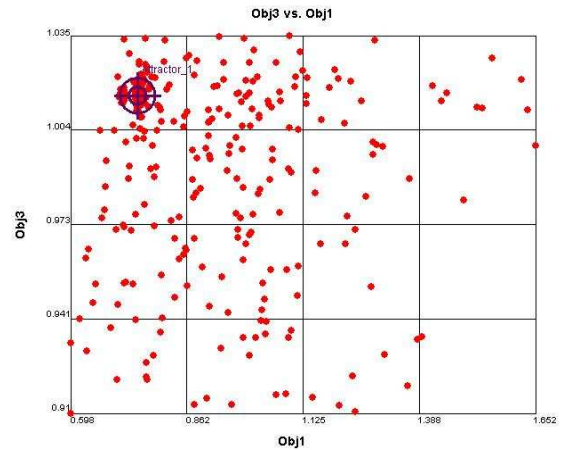
Figure 2. Example of Design Space Sampler

2) *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.



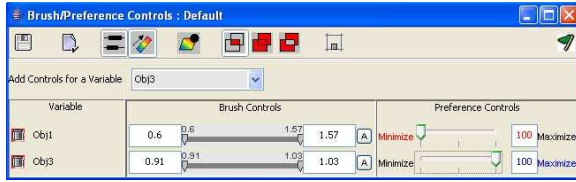
(a) 100 initial samples



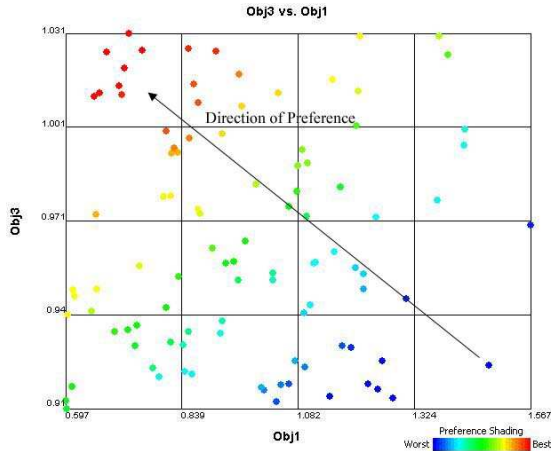
(b) New samples generated near attractor

Figure 3. Example of Point Sampler (Attractor)

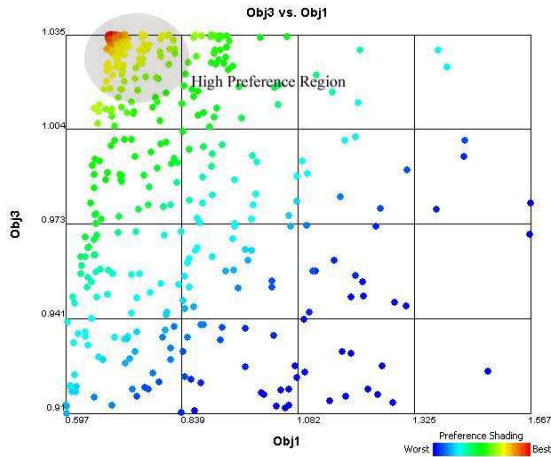
3) *Preference-based samplers* allow users to populate the trade space in regions that perform well with respect to a user-defined preference function. New sample points are also generated by the DE algorithm, but the fitness of each sample is defined by the user's preference structure instead of the 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



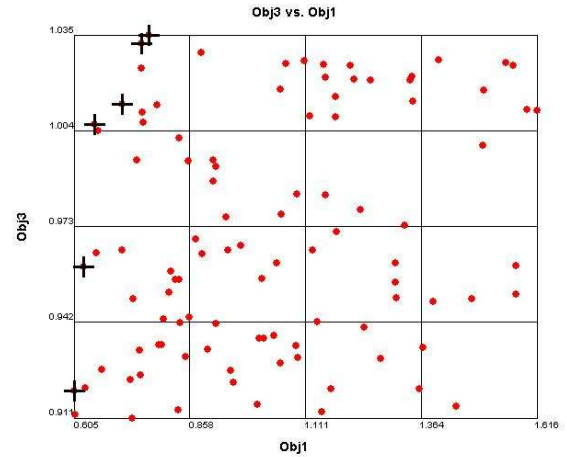
(c) New samples generated in direction of preference

Figure 4. Example of Preference-based Sampler

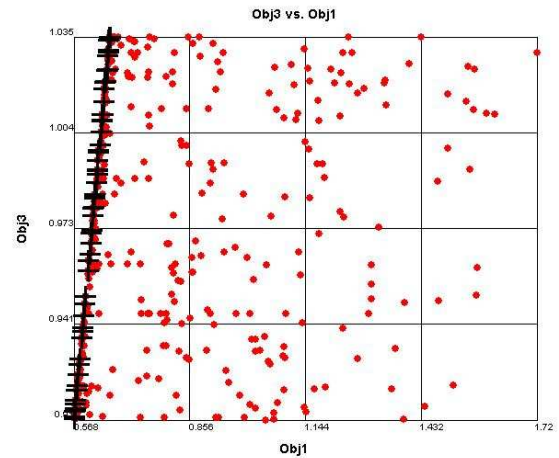
4) *Pareto samplers* are used to bias the sampling of new designs in search of the Pareto frontier once the user has defined his/her preferences on the objectives. The DE algorithm is again used to accomplish this sampling but is modified to solve multi-objective problems [20]. An example of this sampler is shown in Figure 5. Using the same preference (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.

These visual steering commands can be used together in any combination to explore the trade space. When used in concert with the 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 next section describes a study that compares different combinations of these visual steering commands to a multi-objective genetic algorithm that is executed on the same problem with no human intervention.



(a) Initial samples (Pareto points denoted by +)



(b) New samples generated along Pareto frontier

Figure 5. Example of Pareto Sampler

4 EXPERIMENTAL SET-UP AND USER TRIALS

4.1 Test Problem

The test problem used in this study is a vehicle configuration model that was developed to evaluate the technical feasibility of new vehicle concepts [21,22,23]. Table 1 summarizes the problem definition that is used for this trade space exploration 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, 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, we want Obj1 to be smaller than the baseline value while larger values are better for the other four objectives. While stating these very general preferences beforehand may seem counter-intuitive to trade space exploration, the end goal is to demonstrate that the visual steering commands used in conjunction with ATSV are more effective at obtaining an equally desirable Pareto frontier than by simply allowing a MOGA to run "blindly".

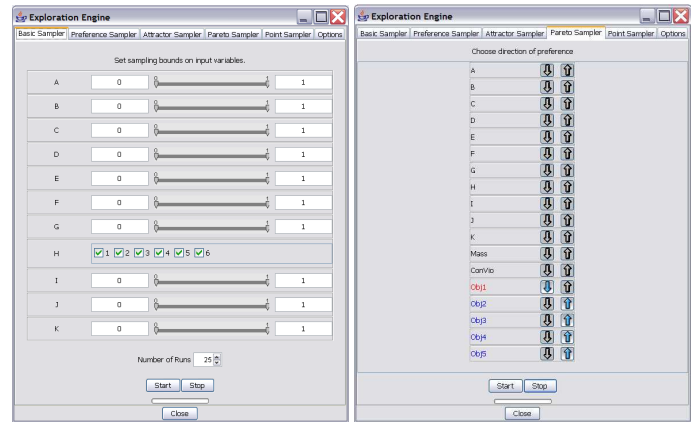
Table 1. Vehicle Problem Definition

| 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 |

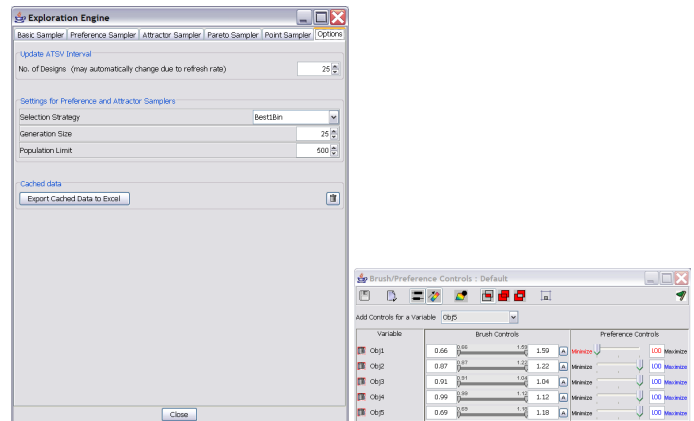
4.2 Description of User Trials

Two sets of user trials were defined for the study based on the allocated number of function evaluations that could be used: ~5,000 and ~10,000, and two users performed each set of trials to account for any randomness in the algorithms, placement of attractors, or specification of brush/preferences controls. While there are nearly an infinite number of combinations of brushing, preference controls, and visual steering commands that could be implemented in ATSV, we allowed a more experienced user to step through a process that felt "natural" and then had the less experienced user replicate those steps as accurately as possible. The experienced user was asked to do this multiple times, creating four different combinations (Trials 1-4) that used approximately 5000 function evaluations and four different combinations (Trials 5-8) that used approximately 10,000 function evaluations.

The ATSV set-up and parameter settings for Trials 1-4 are shown in Figure 6. Table A in the Appendix A describes the specific combinations of brush/preference controls and visual steering commands used for each of Trials 1-4. All four trials begin with a relatively small set of randomly generated samples before proceeding to different combinations of samplers. Initial motivation for setting attractors comes from the fact that many designers use pair-wise comparisons in making decisions [24]; comparing only two objectives makes it easy to see relationships among them. Not all attractors were placed for this reason; others were placed in an attempt to fill in gaps (similar to how the Gap Analyzer was used [23]) in the Pareto frontier, or to push the frontier toward optimality as the trade space exploration process unfolded. The preference and Pareto samplers were also used in an attempt to fill in the Pareto frontier. Unless specified, the Exploration Engine options (see Figure 6c) were left at default settings of generation size = 25, population limit = 500, and the Best1Bin selection strategy. Figure 7 shows the Pareto frontiers obtained by both users after performing Trial 4.

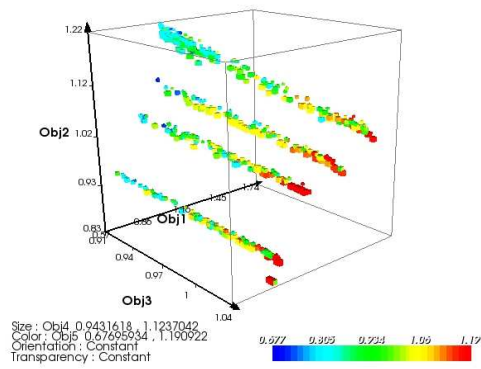


(a) Basic sampler (b) Preferences for Pareto sampler

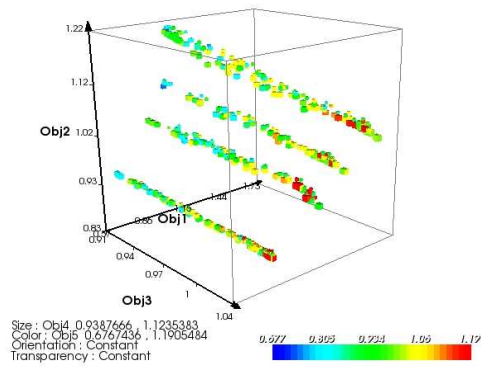


(c) Specifications in options tab (d) Preference control settings

Figure 6. ATSV Set-up for Trials 1-4



(a) Trial 4 User 1 Pareto frontier



(b) Trial 4 User 2 Pareto frontier

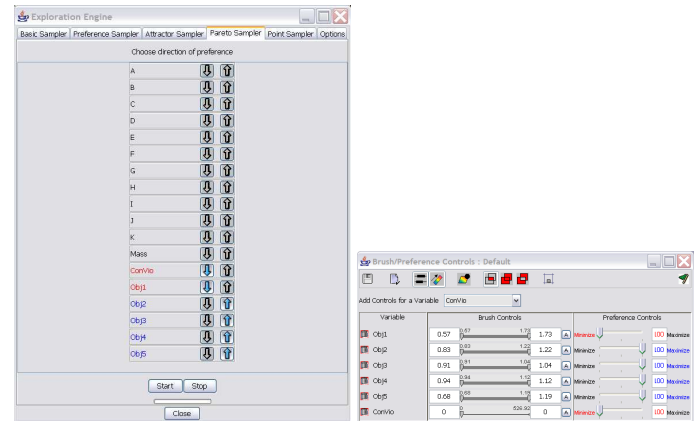
Figure 7. Example of Results from Trial 4

The second set of four trials (Trials 5-8) each used approximately 10,000 points, doubling the number of function evaluations allocated to the user. These four trials all began with a small set of random samples to allow the user to specify preferences (see Figure 8), but they then varied widely in the order and type of attractors and samplers used. Table B in the Appendix describes the user preference settings and specific combination of visual steering commands that were used by each user for Trials 5-8. 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 set it halfway through the trial. Unless specified, the same options and parameter settings were used for these trials as Trials 1-4 (see Figure 6c). Figure 9 shows an example of the Pareto frontiers that the two users obtained after completing Trial 6.

Major attributes of each of the trials can be seen as follows:

- Trial 1 – Attractors placed based on 2 objective interactions
- Trial 2 – Tried to push frontier based on what is visible
- Trial 3 – Attractors placed based on 3 objective interactions
- Trial 4 – Similar to Trial 2 with options changed to allow for more attractors
- Trial 5 – Tried to push frontier beyond what is visible

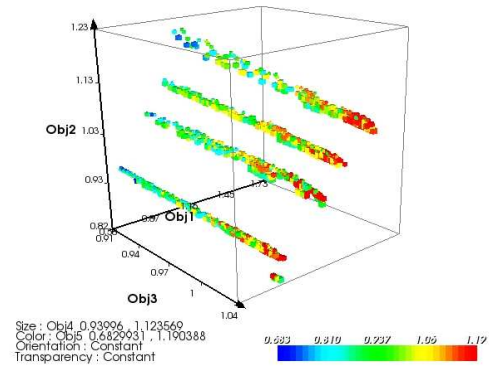
- Trial 6 – Tried to fill in frontier with 3 objective interactions
 - Trial 7 – Allow Pareto and Preference-based samplers to alternate and move through feasible space before restarting
 - Trial 8 – Similar to Trial 7 with a different selection strategy
- These eight trials are based mainly on what felt “natural” to the users and represent only a fraction of the possible combinations.



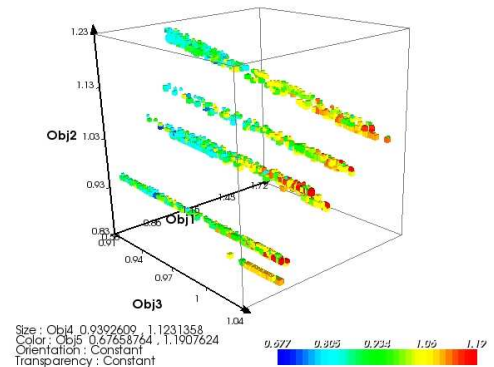
(a) Preferences for Pareto sampler

(b) Preferences control settings

Figure 8. ATSV Set-up for Trials 5-8



(a) Trial 6 User 1 Pareto frontier



(b) Trial 6 User 2 Pareto frontier

Figure 9. Example of Results from Trial 6

4.3 Reference Pareto Set

For comparison purposes, the reference (or “best known”) Pareto frontier comes from an exhaustive multi-objective GA (MOGA) search that was performed previously on the same test problem by its originators [22]. 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 developed that would direct the MOGA to find designs in those areas if such a region was found [23]. The exhaustive MOGA used approximately 80,000 function evaluations to create this reference Pareto frontier (44,769 points in the final population, 5,561 are Pareto-optimal over the continuous objective function space). Even with the Gap Analyzer, the MOGA ran “blindly”, requiring no human intervention while searching the trade space; hence, it provides a suitable benchmark for this study.

4.4 Performance Measures

To quantify the performance and compare the results of the genetic algorithms rigorously, a variety of performance metrics have been developed [25]. Okabe, et al. [26] 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 [27] proposed a hyper-volume metric, which evaluates the size of the dominated space. Reed and Tang [28] 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 relative computational efficiency, accuracy, and ease-of-use. The ϵ -performance metric developed by Kollat and Reed [29,30] 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 of reference set solutions that are found by the GA within this level of precision is reported as 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 in this study 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 frontier obtained from the exhaustive MOGA that used 80,000 function evaluations.

5 ANALYSIS AND DISCUSSION OF RESULTS

Figure 10 provides a visual comparison of the resulting Pareto frontiers from individual trials along with the reference

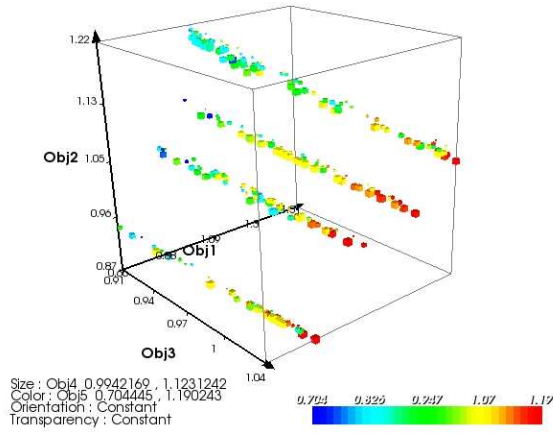
set. While it is difficult to make comparisons in 5-D, we can identify from these figures the trials that did well and those that did not. For instance, Trial 4 by User 2 (see Figure 10b) is much sparser than the other figures, especially when compared to the reference set (see Figure 10e). Figure 10f provides a composite of all eight trials where the reference set from the exhaustive MOGA is shown in blue, solutions from Trials 1-4 and Trials 5-8 that are the same as the MOGA solutions are shown in green and red, respectively. As expected, the MOGA solutions dominate the majority of the solutions obtained from either set of trials; however, it is promising to see that some solutions remain given that the trials used about 5000 and 10,000 function evaluations compared to MOGA’s 80,000.

Before comparing the sets of solutions quantitatively using the ϵ -performance metric, we need to determine a suitable value for epsilon. After confirming that all input and output variables were normalized by the same ranges and scaled against the same baseline values, we computed the differences between the objectives of every pair of designs in the reference set from MOGA. We found that the smallest difference between any two designs was so close to zero that any reasonable value of epsilon could be selected. While choosing an epsilon value that was too large would reduce each set to the point that comparison would be meaningless, choosing an epsilon value that was too small would make it almost impossible to find designs within one epsilon of each other in each objective given that it is a 5-D space. Therefore, after performing a sensitivity study of epsilon values between 0.001 and 0.1, a value of 0.01 was selected for each objective and used for this analysis.

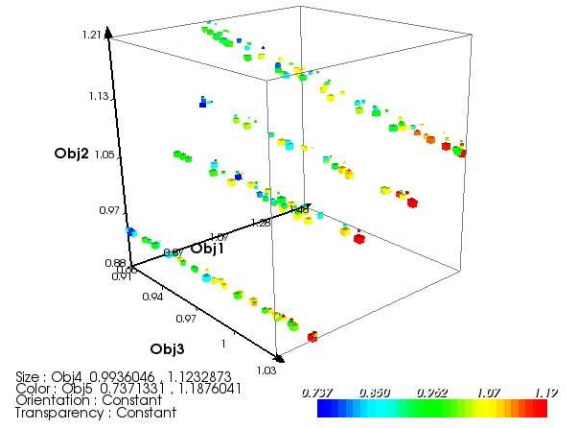
Table 2 shows the results of each trial using the ϵ -performance metric for both users (v1 and v2). As discussed in Section 4.4, the reported results are obtained by comparing each user’s resulting Pareto set from each trial to the reference set obtained from the exhaustive MOGA, which is the best approximation of the “true” Pareto frontier that we can obtain. We see that when using only 5,000 points, both users are able to obtain 9.3% - 13.9% of the reference set; when allocated 10,000 points, both users are able to increase this range to 12.9% - 22.4%. Thus, users are able to obtain, on average, 12% and 18% of the solutions on the Pareto frontier by using 1/16th and 1/8th, respectively, of the number of function evaluations used by the exhaustive MOGA.

Table 2. Results based on ϵ -performance Metric

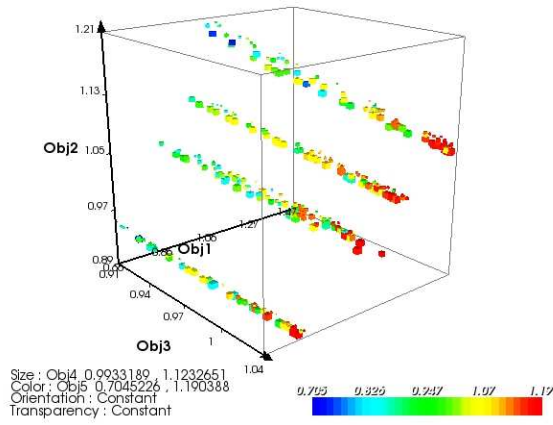
| Trials 1-4 (5000 Points), % | | | | | | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Trial | 1 v1 | 1 v2 | 2 v1 | 2 v2 | 3 v1 | 3 v2 | 4 v1 | 4 v2 | Avg |
| found | 0.00 | 0.61 | 0.40 | 0.20 | 0.40 | 1.01 | 0.20 | 0.61 | 0.43 |
| dominating | 12.90 | 10.69 | 9.68 | 13.71 | 11.69 | 12.90 | 13.71 | 8.67 | 11.74 |
| Total | 12.90 | 11.29 | 10.08 | 13.91 | 12.10 | 13.91 | 13.91 | 9.27 | 12.17 |
| Trials 5-8 (10,000 Points), % | | | | | | | | | |
| Trial | 5 v1 | 5 v2 | 6 v1 | 6 v2 | 7 v1 | 7 v2 | 8 v1 | 8 v2 | Avg |
| found | 0.40 | 0.00 | 0.61 | 0.20 | 0.40 | 0.61 | 0.20 | 1.21 | 0.45 |
| dominating | 14.31 | 12.90 | 21.37 | 22.18 | 17.34 | 19.36 | 17.94 | 18.15 | 17.94 |
| Total | 14.72 | 12.90 | 21.98 | 22.38 | 17.74 | 19.97 | 18.15 | 19.36 | 18.40 |



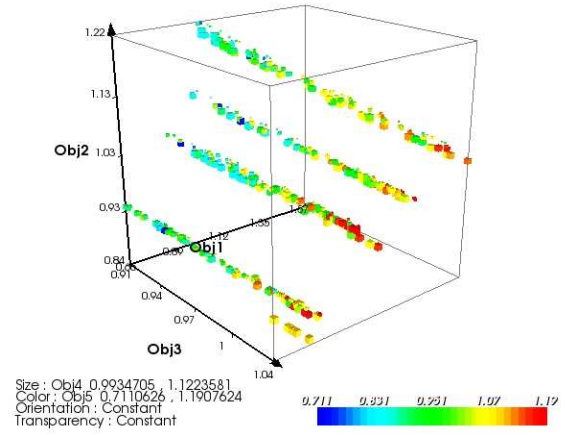
(a) Trial 4 User 1 Pareto frontier



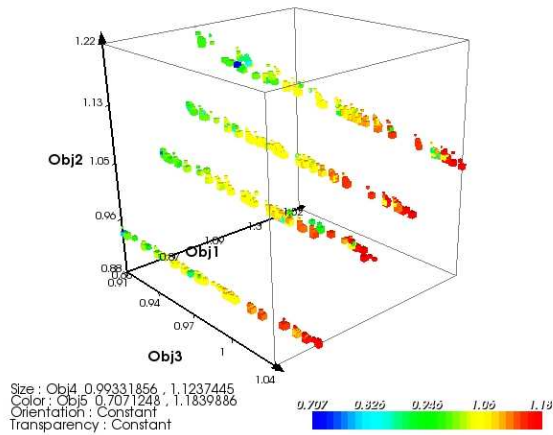
(b) Trial 4 User 2 Pareto frontier



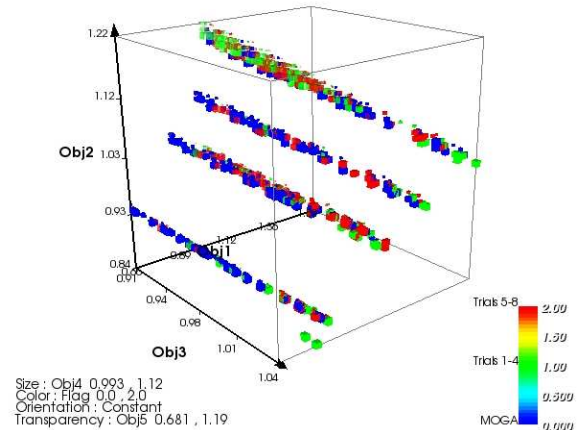
(c) Trial 6 User 1 Pareto frontier



(d) Trial 6 User 2 Pareto frontier



(e) Reference Pareto frontier



(f) Pareto solutions color-coded by trial

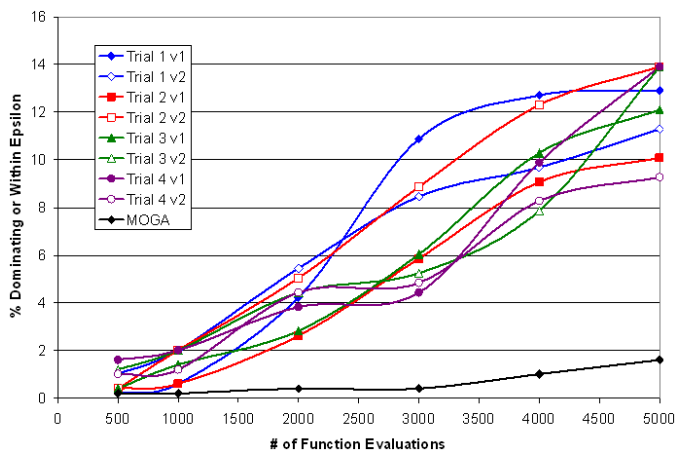
Figure 10. Example of Visual Comparisons of Resulting Pareto Frontiers

Table 2 also shows that the 10,000 function evaluation trials perform better than the 5,000 function evaluation trials as one would expect, with Trial 6 performing the best. Not surprisingly, the percentage of designs found within epsilon of

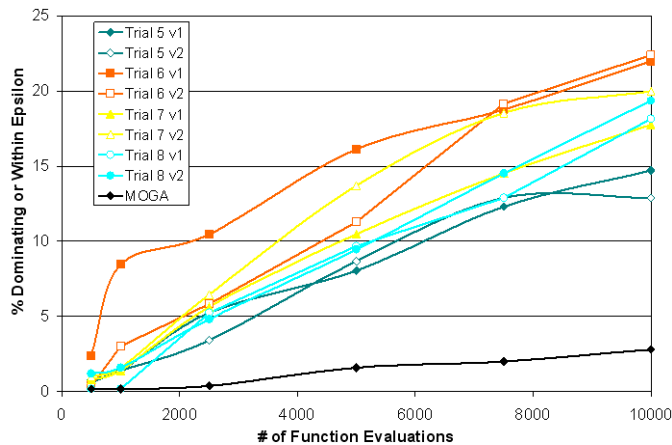
the reference solutions is very low for every trial. This is likely a result of the objective space being 5-D, which makes it very difficult to find two designs that fall within 0.01 of each other in all five objectives. An unexpected result, however, is how high

the percentage of designs dominating the reference is for each trial. This indicates that the reference set generated by the exhaustive MOGA is likely not the “true” Pareto set, but rather itself an approximation of the “true” set. This also shows that a user-guided trial in ATSV could possibly have a better chance of obtaining the “true” Pareto set than the MOGA.

To gain more insight into the performance of each trial as well as the evolution of solutions toward the Pareto frontier, we plot the ϵ -performance metric at a series of intervals leading up to the allocated number of function evaluations. In particular, Figure 11a shows the performance of each user (v1 and v2) in Trials 1-4 at 500, 1000, 2000, 3000, 4000, and 5000 function evaluations; Figure 11b shows a similar progression for each user for Trials 5-8 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.



(a) Results from Trials 1-4 and Exhaustive MOGA Search



(b) Results from Trials 5-8 and Exhaustive MOGA Search

Figure 11. Evolution of Pareto Frontiers in Each Trial and the Exhaustive MOGA Search

While the results from Table 2 may not have been too convincing, Figure 11 clearly illustrates the benefit of having the user “in-the-loop” during the optimization process. In all trials, both users have substantially out-performed 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 11a, the MOGA has obtained fewer than 2% of the Pareto frontier in its first 5000 function evaluations compared to 9.3% - 13.9% in Trials 1-4. Likewise, even when the number of function evaluations has doubled to 10,000, the MOGA has still found fewer than 3% of the Pareto frontier solutions compared to the 12.9% - 22.4% obtained in Trials 5-8 as indicated in Figure 11b. In both cases, this represents a 4x -7x 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 consistent, regardless of the combination of visual steering commands that are used or the designer implementing them.

6 CONCLUSIONS AND FUTURE WORK

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 – regardless of the combination in which they are invoked – can provide a 4x -7x increase in the number of Pareto solutions that are obtained when the human is “in-the-loop” during the optimization process. As such, this study provides the first empirical evidence of the benefits that interactive visualization-based strategies can provide in support of engineering design optimization and decision-making.

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 goodness of the Pareto frontiers more thoroughly but also providing guidance to users if they were computed in real-time during the trade space exploration process. A more extensive reference set should also be developed given the high dimensionality of the trade space that is being explored. Having a more complete reference set would prevent any individual trial from being able to dominate the reference set (or any portion thereof); trials would only be able to find points in the reference set within epsilon. Finally, the study should also be repeated with test problems of different sizes and complexity as well as with users with different levels of experience to demonstrate how widely applicable – and beneficial – the trade space exploration process is.

ACKNOWLEDGMENTS

We thank Scott Ferguson for his assistance with the vehicle configuration model and the results from the exhaustive MOGA. This work has been supported by the National Science Foundation under Grant No. CMMI-0620948. Any opinions,

findings, and conclusions or recommendations presented in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

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] Simpson, T. W., Donndelinger, J. A., Yukish, M. and Stump, G., 2007, "Visual Steering Commands for Trade Space Exploration: User-Guided Sampling with Example", *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Las Vegas, NV, DETC2007/DAC-34684.
- [5] 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.
- [6] 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.
- [7] Messac, A. and Chen, X., 2000, "Visualizing the Optimization Process in Real-Time Using Physical Programming," *Engineering Optimization*, 32(6), 721-747.
- [8] Winer, E. H. and Bloebaum, C. L., 2001, "Visual Design Steering for Optimization Solution Improvement," *Structural Optimization*, 22(3), 219-229.
- [9] 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.
- [10] Scott, S. D., Lesh, N. and Klau, G. W., 2001, "Investigating Human-Computer Optimization", *CHI*, Minneapolis, MN.
- [11] Miettinen, K. and Makela, M. M., 2006, "Synchronous approach in interactive multiobjective optimization," *European Journal of Operational Research*, 170, 909-922.
- [12] Madar, J., Abonyi, J. and Szeifert, F., 2005, "Interactive Particle Swarm Optimization", *International Conference on Intelligent Systems Design and Applications*, IEEE.
- [13] Michalek, J. and Papalambros, P., 2002, "Interactive Design Optimization of Architectural Layouts," *Engineering Optimization*, 34(5), 485-501.
- [14] Stump, G., Yukish, M. and Simpson, T. W., 2004, "The Advanced Trade Space Visualizer: An Engineering Decision-Making Tool", *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, AIAA-2004-4568.
- [15] 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.
- [16] 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.
- [17] Becker, R. A. and Cleveland, W. S., 1987, "Brushing Scatterplots," *Technometrics*, 29(1), 127-142.
- [18] 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.
- [19] Price, K., Storn, R. and Lampinen, J., 2005, *Differential Evolution - A Practical Approach to Global Optimization*, Berlin, Springer.
- [20] Robic, T. and Filipic, B., 2005, "DEMO: Differential Evolution for Multiobjective Optimization." *Third International Conference on Evolutionary Multi-Criterion Optimization*, Guanajuato, Mexico, Springer, 520-533.
- [21] 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-2006-7056.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] Okabe, T., Jin, Y. and Sendhoff, B., 2003, "A Critical Survey of Performance Indices for Multi-Objective Optimisation," *Evolutionary Computation*, 2, 878-885.
- [27] Zitzler, E., 1999, *Evolutionary Algorithms for Multiobjective: Methods and Applications*, Zuerich, Switzerland, Swiss Federal Institute of Technology.
- [28] Tang, Y., Reed, P. M. and Wagener, T., 2005, "How Effective and Efficient are Multiobjective Evolutionary Algorithms at Hydrologic Model Calibration," *Hydrology and Earth System Sciences Discussions*, 2, 2465-2520.
- [29] 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.
- [30] 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.

APPENDIX

Table A. Specification of Visual Steering Commands for Trials 1-4

| Trial 1 (Total Points: 5025) | Trial 2 (Total Points: 5075) |
|--|---|
| <ul style="list-style-type: none"> - 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 | <ul style="list-style-type: none"> - 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 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]) |
| Trial 3 (Total Points: 5525) | Trial 4 (Total Points: 5375) |
| <ul style="list-style-type: none"> - 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 | <ul style="list-style-type: none"> - 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): Set objective 3 at current limit of the scatter plot window - Point attractors: Set at current limits of the scatter plot window (on objectives [3 & 4], [4 & 5]) |

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

| Trial 5 (Total Points: 10,325) | Trial 6 (Total Points: 10,075) |
|--|--|
| <ul style="list-style-type: none"> - 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 $\pm 5\%$ 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 | <ul style="list-style-type: none"> - 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: Use these specific values to fill in the Pareto frontier ([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 |
| Trial 7 (Total Points: 10,125) | Trial 8 (Total Points: 10,275) |
| <ul style="list-style-type: none"> - 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 | <ul style="list-style-type: none"> - 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 |