Proceedings of DETC'02
ASME 2002 Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
Montreal, Canada, September 29-October 2, 2002

DETC2002/DAC-34130

VISUALIZATION OF MULTIDIMENSIONAL DESIGN AND OPTIMIZATION DATA USING CLOUD VISUALIZATION

John Eddy
Graduate Research Assistant
Department of Mechanical and Aerospace Engineering
University at Buffalo
Buffalo, NY 14260
johneddy@eng.buffalo.edu

Kemper E. Lewis[1]
Associate Professor
Department of Mechanical and Aerospace Engineering
University at Buffalo
Buffalo, NY 14260
kelewis@eng.buffalo.edu

Abstract
As our ability to generate more and more data for increasingly large engineering models improves, the need for methods for managing that data becomes greater. Information management from a decision-making perspective involves being able to capture and represent significant information to a designer so that they can make effective and efficient decisions. However, most visualization techniques used in engineering, such as graphs and charts, are limited to two-dimensional representations and at most three-dimensional representations. In this paper, we present a new visualization technique to capture and represent engineering information in a multidimensional context. The new technique, Cloud Visualization, is based upon representing sets of points as clouds in both the design and performance spaces. The technique is applicable to both single and multiobjective optimization problems and the relevant issues with each type of problem are discussed. A multiobjective case study is presented to demonstrate the application and usefulness of the Cloud Visualization techniques.

1 Motivation
Engineers are increasingly able to generate more and more data for large system models, and as a result, the need for methods for managing and visualizing that data becomes greater. From a design perspective, large-scale analysis can result in a huge number of potential design configurations existing in complicated multi-dimensional spaces. This information could be presented to a designer as page after page of printed data. However, it is very difficult to make sense of large data structures in print form.

Some of the first attempts at using visualization methods to aid decisions in design and optimization are found in [1]. More recent advances in computer visualization and Virtual Reality (VR) [2-4] are allowing designers and scientists to interact and manipulate vast amounts of data. Until these innovations, computers were solely relied on to interpret results and compute answers based on programs written. It is now possible to interact with these large datasets even while they are being used in running analyses [5-11]. Users have the ability to compress large amounts of data into a visual format, to investigate trends and relationships that could not be seen otherwise, and then make informed decisions regarding a product or process design.

Commercial companies such as Raytheon [12] and Boeing [3], among others, have attempted to improve their own design processes by taking advantage of Virtual Reality and Scientific Visualization. Virtual Reality and Scientific Visualization offer methods and concepts to produce graphical representations (pictures, graphs, etc.) of complex data. The enormous computing power readily available today has made these technologies more and more useful in recent years. Tools and techniques based on these technologies are being developed that can further improve the efficiency and accuracy of the solutions to these complex designs. However, there is still a need for process improvement and multidimensional data representation, as well as for better incorporation of heuristics and design knowledge. One approach is through the implementation of Computational Steering concepts. Computational Steering [7-10] is the

[1] ASME Member, Corresponding Author

implementation of Virtual Reality and Scientific Visualization into a process or analysis, so as to give a researcher or practitioner the ability to view how a solution procedure is progressing. The researcher has the ability to alter parameters while the analysis is running to interactively "steer" it to a solution. This ability to interact visually with design and optimization processes has the potential to benefit decision making greatly, as estimations indicate that approximately 70% of a humans attention is dedicated to visual input [13]. To this end, visualization can and should be considered a solution tool rather than simply a way to present results [14].

Computational steering is the paradigm adopted in this work, as the focus of this paper is on presenting an approach to representing multidimensional design optimization problems to facilitate the decision making and steering capabilities of a designer. The techniques presented in this paper are applicable to both single and multiobjective optimization problems. Because of the significant dependence that engineers have on visual cues and information representation, it seems sensible to provide visually enhanced design steering and optimization methodologies. Easily interpreted visual cues such as color, shape, relative size, etc. can be used effectively to convey trends or large amounts of perhaps imprecise information quickly. They can also be used to add dimension to data beyond the three spatial dimensions. In this paper, some of these cues are integrated into a visualization framework, which acts as the designer interface while single or multiobjective optimization techniques operate at a lower, processing level. This framework not only provides insight into the nature of the problem and the optimization algorithm, but also helps prevent wasted analysis and supplies the designer with an opportunity to interact with the analysis.

Real-time visualization methods can be classified into two general categories: Artifact-Based Visualization (ABV) schemes and Non-Artifact-Based Visualization (Non-ABV) schemes [11]. Artifact-based schemes are those that are typically imposed on a physical object with a prescribed geometry, such as representing the stress in a beam by color contours imposed on the beams geometry. Non-Artifact-based schemes are those that are not constrained by a physical geometry such as visualization of process optimizations.

Non-ABV can also be further divided into two subcategories [11]. The first includes those methods that are based on a topographical view of the design space whereby a fit is made to the performance objectives based on the results of many a priori system evaluations. The results are usually viewed as contour plots. As pointed out in [11], approaches of this type require that system evaluations be performed before the optimization begins in order to create the contours, they do not accommodate multi-dimensional data very well, and the representation may not be meaningful due to inadequacy of the system model and/or approximations used to generate the contour plots.

The second subcategory contains non-ABV methods that do not require a topographical representation of the problem, such as tabular data, 2D plots, parallel coordinate based, and physical programming based methods. The method presented here, Cloud Visualization, is a category 2b visualization method meaning that it is a Non-ABV method that does not begin with a topographical view of the design space. Readers interested in more detailed discussion of the advantages and disadvantages of ABV and non-ABV are directed to [11] for more information.

The visualization procedure discussed in this paper is focused primarily on optimization problems, both single and multiple objective. There are various challenges in both types of problems that visualization techniques can be used to solve and in the next section, some of these general challenges of engineering optimization are presented.

2 Engineering Optimization Challenges

In this section, both single objective and multiobjective decision problems are discussed. It is these types of problems that are the focus of the visualization approaches presented in this paper.

2.1 Single Objective Design Problems

Single objective problems are defined by a single measure of performance that may be a single criteria or a combination of many. The challenges in single objective optimization involve primarily being able to find and represent solutions to the problem. First, finding the globally optimal solution and being able to verify the level of optimality is a challenging task. Second, for effective decision-making, areas with equivalent locally optimal solutions may exist and must be found. Third, being able to represent multivariable space effectively in order to support the first and second challenges is paramount. Each of these challenges can be influenced and supported by visualization techniques coupled with numerical techniques.

2.2 Multi-Objective Design Problems

Many designers concede that there is typically more than one criterion that must be considered when choosing a design configuration. Often, a large system is decomposed into smaller subsystems each having its own set of objectives, constraints, and parameters. The performance of the final design is a function of the performances of the individual subsystems. It then becomes necessary to consider the tradeoffs that occur in a multi-objective design problem.

The objectives in a multi-objective problem may relate differently to one another. Two objectives may be in competition, meaning that improvement of one typically comes at the expense of the other. Objectives may also be in cooperation with each other, meaning that improvement of one typically accompanies improvement of the other. In this case, there is a single superior design and tradeoffs are not necessary. The third possibility is that there is no relationship, as is the case if the objectives have no variables in common.

The case in which objectives are in competition with one another is the most interesting. This situation gives rise to the concept of Pareto optimality. For such problems, the goodness of a design is a function of its Pareto dominance. A point, $\bar{x}'$, is Pareto dominant if and only if there is no feasible design variable vector, $\bar{x}$, for which:

$$f_i(\bar{x}) \le f_i(\bar{x}') \quad \text{for all i, i =1,}n \qquad (1)$$
$$f_i(\bar{x}) < f_i(\bar{x}') \quad \text{for at least one i, } 1 \le \text{i} \le n$$

where n is the number of objectives to be minimized.

There are three primary challenges in multiobjective design problems. First, the generation of the Pareto set is a challenging problem [15-16]. Second, selecting a solution from among a Pareto set is a task made more challenging with the existence of distinct and dynamic preferences among designers [17]. These preferences are typically centered on the system objectives (how the system performs). However, in order to actually produce a system that behaves in a desired way, values for design variables must be chosen. The third challenge deals with the relationship between the objectives and design variables. The task of mapping a chosen Pareto solution into a set of design variable values is a one-to-many mapping problem that can be extremely challenging [18]. The trade-offs that result between competing objectives are very difficult to perceive from printed data. Visual aids can greatly improve a designer's understanding of the relationships developing among the objectives and between the objectives and design variables. The solution of these challenges can be facilitated using effective visualization technologies and tools and is the focus of this paper.

In the next section, the foundations for the visualization techniques and technologies that are utilized to help meet these computational steering challenges in both single and multiobjective optimization problems are presented.

3 Background

Cloud Visualization is a means by which a designer can view all previously generated design information in both the design space and the performance space simultaneously. Its implementation resembles that of the Visiview method developed in [19] with some significant differences. The design space is defined by the design variables of the problem while the performance space is defined by the performance objectives. The two spaces are displayed in separate windows that can be linked. Some general information regarding Cloud Visualization is presented in the next section. Then in the following sections, specific details for both single and multiple objectives optimization implementations are discussed.

3.1 General Implementation

All the graphics presented in this paper were created using openGL and implemented on a machine with a 600 MHz Pentium 3 processor.

Design information is presented as a cloud of design points plotted along three or less axes that represent either design variables or objective functions. The cloud is presented in different colors to reflect the properties of certain portions. The data represented may be single or multi-objective.

Use of Color

The use of color in visualization techniques can be very effective, but it can also be overwhelming, decreasing the effectiveness of the presentation and the value of the information [20]. The use of color in visualization should not create "graphical puzzles" for which the user is trying to remember what the color representations imply instead of interpreting the information in the chart or graph and making a rational decision [21]. Further, too many colors can only confuse users and the simple use of gray scales can often be more effective than elaborate color schemes [21].

To this end, two color schemes are available in Cloud Visualization, depending upon what type of decision is to be made. The first colors the designs in a rainbow according to their goodness such that the best design(s) are red and the worst are violet. Determination of a designs' goodness is discussed in Sections 2.1 and 2.2. All infeasible designs appear in gray. This scheme allows the user to quickly follow the progression of color to good areas of the space.

For some datasets, particularly in multi-dimensional spaces, these trends do not appear in a sensible order when looking at only 3 dimensions (at-a-time). Therefore, the second color scheme colors infeasible designs gray, feasible designs green, and the best design(s) red. In the case of multi-objective optimization, yellow is used for points that are *distinct* Pareto points. To compute distinct Pareto points, the design space is divided into a hyper-grid. A *distinct point* is one that exists *alone* in a grid location (see [22] for more details). An example of each color scheme is provided in Figure 1, which represents a multi-objective problem for which the lower right corner of the space is the desirable region.

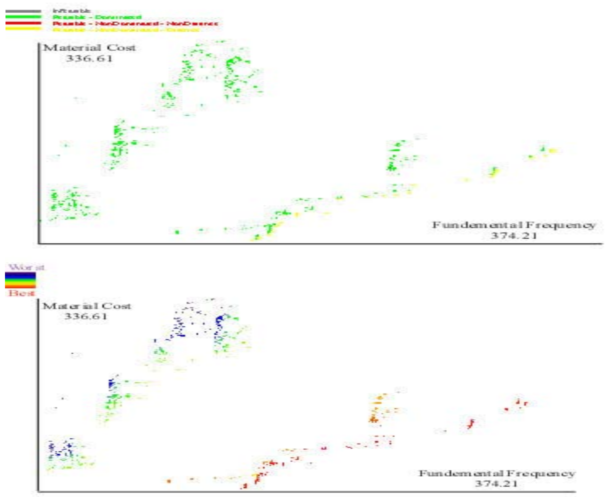

Figure 1: Color Schemes (Blocked color on top, Rainbow on bottom)

The color schemes enable a designer to focus in on favorably colored areas of the spaces. In order to isolate desired regions, the user can select individual points or groups of points that will then be displayed by themselves. When selecting a group of points, the user has the option to rescale them in color with respect to each other to further focus in on good areas of the space. This can be done in a window showing the design variable space or performance space and the color will be updated in all associated windows. An example of this capability is presented in Figure 2.

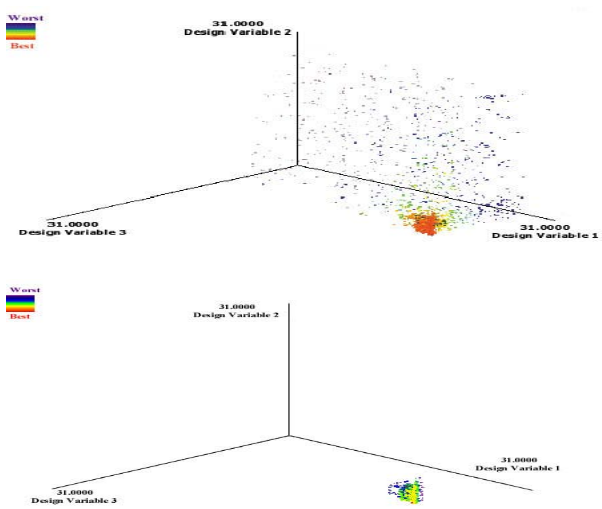

Figure 2: Images representing color rescaling

The figure on the top shows all feasible points developed at some point during the solution of a single objective design problem. The best points are red and it is clear where the desirable region of the space is. However, there are clearly a large number of designs in that region and the colors are too similar to distinguish them from one another. Therefore, in the image on the bottom, only the points in the desired region are selected and they have been rescaled in color relative to one another.

The user can now further focus in on the good region of the space by applying some of the interactive capabilities described in the following and making further selections.

Interactive Capabilities

To further aid exploration, the user can dynamically rotate the spaces around all three spatial axes, translate the spaces along all three axes, and zoom into any area of the spaces while in any orientation. Additionally, the designer can invoke cutting planes and view 2-D cross sections of the cloud to aid in point selection as shown in Figure 3.

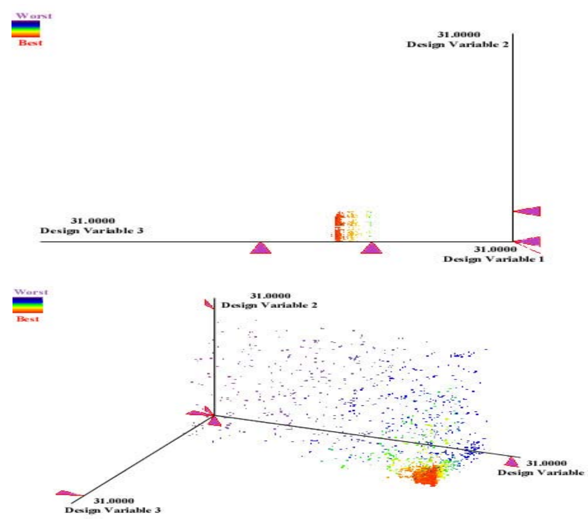

Figure 3: Images demonstrating cutting planes

Finally, the user can select groups of points with different attributes from pop-up menus for viewing. The different groups available for viewing depend on the type of optimization and are discussed in detail in further sections. As an example, the user could select to see only the feasible points and then all infeasible points would be hidden.

Design Steering

As described in the motivation section, it is desirable to interact with an optimization tool as it is running. The current implementation of Cloud Visualization does allow for some relatively simple interactions. While in the design space, the user can choose to relocate a point by dragging it to a new location. The point is not re-evaluated immediately but instead is colored black and labeled invalid. Upon re-entering the optimization run, the point is re-evaluated and considered in its new location. While in either space, the user can select groups of points and label them desirable or undesirable. This capability is especially useful when using methods like

Genetic Algorithms in which large groups of points are considered simultaneously. In this case, upon returning to the optimization, the points can be given special consideration depending on whether they are labeled desirable or undesirable. Figure 4 shows a capture of the design points at some time during the run of a multi-objective genetic algorithm (MOGA). The user has selected the cyan points as undesirable and upon returning to the algorithm, they will be removed from the population allowing the remaining points to continue.

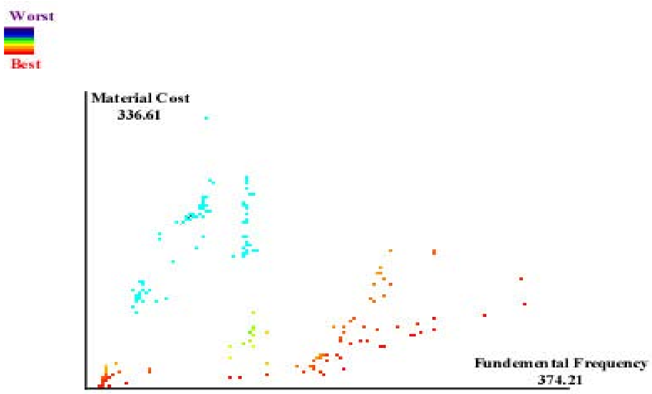

Figure 4: Selection of points for design steering in Cloud Visualization

Handling Multiple Dimensions

In the case of more than three design variables or objective functions, a single point in a 3D space might be selected that actually represents multiple unique designs. In this case, another window is created with three or less new design variables or objective functions (depending on how many remain unused) while all previously used variables or functions are held constant in the new window. The initial window is the parent of the resulting window. The points in the parent window are then locked and cannot be changed unless the child window is destroyed. The windows can be cascaded out until all the variables or objectives are represented on a plot as demonstrated in Figure 5.

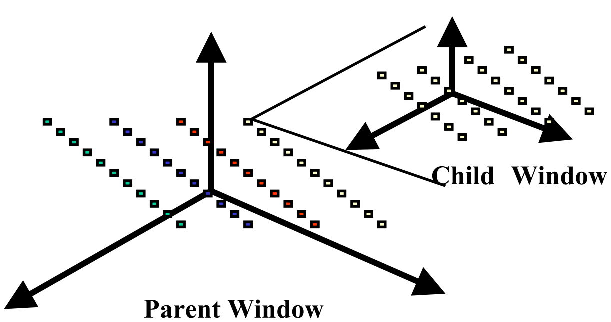

Figure 5: Point selection for multiple variables or objectives

Mapping Between Spaces

In order to be an effective optimization tool, Cloud Visualization must address the issue of mapping points between the performance and design space. For a given set of equations describing a system, there is a one-to-one mapping from the design space into the performance space. That is, given some set of design parameters, the performance can be singularly determined according to the objective functions. The opposite is not necessarily true, meaning that given some performance level, there may be many design configurations that will suffice. Mathematically speaking, there are typically more variables (design parameters) than equations (objective functions) in a design problem. Consider this analogy: given some automobile design, the top speed can be singularly determined for that design based on engine horse power, drag coefficient, etc. However, given a desired speed, there may be many automobile designs capable of achieving it. Figure 6 demonstrates this discrepancy where the mapping from the design space to the performance space is one-to-one, while the mapping from the performance space to the design space is a one-to-many. Previous work to solve this problem was effective for two objective problems, but extension to multiple objectives is provided in Cloud Visualization.

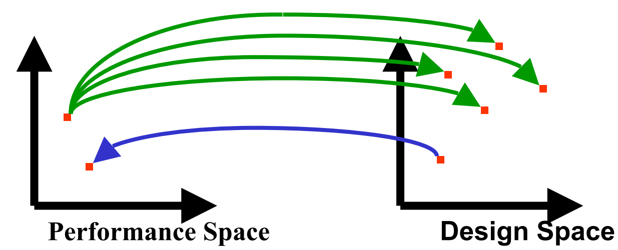

Figure 6: Mapping between design and performance spaces

In Cloud Visualization, if a point is selected in the performance space for which multiple associated design configurations have been found, all associated design points are displayed in any associated design space using the database of points visited during the MOGA population generation. If a hypothetical point is selected in the performance space (not a point on the screen, but a point in space), then a scheme to find all possible design points with the objective function values of the selected point. An optimization problem is formulated for doing the mapping. The problem is stated as shown in Equation (2).

$$
\begin{array}{lll}
\text{Minimize:} & |f_i(\bar{x}) - T_i| & i = 1,n \\
\text{Subject To:} & h_k(\bar{x}) = 0 & k = 1,l \\
 & g_j(\bar{x}) \le 0 & j = 1,m \\
 & x_r^l \le x_r \le x_r^u & r = 1,ndv
\end{array}
\qquad (2)
$$

Where n is the number of objective functions in the original problem formulation, l is the number of equality constraints in the original problem formulation, m is the number of inequality constraints in the original problem formulation, and ndv is the number of design variables. T_i are the objective functions values of the hypothetical point chosen in the performance space and are used as target values. The target values remain constant throughout the optimization run. The objective function values, $f_i(\bar{x})$, change as $\bar{x}$ changes. If the user has selected a point in the performance space that is not physically obtainable (e.g., a point that is better than any Pareto point), this formulation will provide a solution as close to the target performance as possible. If the target values are obtainable then this formulation can provide the desired design vector $(\bar{x})$. In addition, by using a genetic algorithm to solve the problem, any number of satisfactory design vectors may be found in a single run.

In this section, the general implementation characteristics of Cloud Visualization are discussed. In the next sections, specific details pertaining to single and multiple objective problems are discussed further.

3.2 Single Objective Problems

Recall that for a single objective problem, goodness is typically determined by the objective function value. In this case, the rainbow color scheme is applied by coloring the best point(s) pure red, the worst point(s) pure violet, and scaling the remaining points according to their objective function value (disregarding infeasible points, which are gray). The Blocked Color scheme colors infeasible points gray, and feasible points green with the exception of the best point(s) which is/are red.

As mentioned previously, the user can select groups of points for viewing. In the case of a single objective problem, the user can choose to see any of the following groups.

1. All Points.
2. Feasible Points Only.
3. Infeasible Points Only.

With color schemes and optimality filters (creating the groups listed above), an effective means of navigating through large amounts of data quickly to find the best design(s) or regions of good performance with the different color schemes is provided. When utilized during a running analysis, this information can be used by a designer to guide the optimization process as demonstrated in Section 4.

3.3 Multiple Objective Problems

Cloud Visualization treats data from multi-objective problems very similarly to that of single objective problems because single objective optimization is simply a special case of multi-objective optimization where n, the number of objective functions, is equal to 1. Therefore, the rainbow color scheme is implemented as follows by Cloud Visualization for multi-objective problems.

The best designs (Pareto optimal) are identified and labeled level 1 solutions. Then, from the remaining points, those that are non-dominated are labeled level 2. This continues until all design points are accounted for. Color is then assigned according to a designs level such that level 1 designs are pure red, highest level designs are pure violet, and all other designs are scaled by level according to the rainbow. This gives a designer an idea of the quality of a design point relative to the Pareto points.

Regarding selection of groups for display, there are more groups available for viewing in multi-objective problems. They are:

1. All Points.
2. Feasible Points Only.
3. Infeasible Points Only.
4. Pareto Points Only.
5. Distinct Pareto Points Only.

A distinct Pareto point, as defined in [22] and in Section 3.1, is one that is not too close to another in the performance space. The threshold for “too close” is user defined.

With the two available color schemes, the user can watch as the non-dominated frontier develops and quickly make assertions on the optimization based on trends and relationships that appear. Good visualization of multi-objective performance data is very helpful when considering the trade-offs occurring between objectives. Cloud Visualization can help a designer consider the effects and sacrifices of changing performance values while considering design variable information at the same time. In the next section an example is used to demonstrate some of the capabilities of Cloud Visualization when applied to a design optimization problem.

4 Case Study - Vibrating Platform

The vibrating platform problem is a 2 objective optimization problem adapted from a problem presented in [23].

4.1 Problem Description

This is a composite beam consisting of 5 total layers that are symmetrical about the centerline. A diagram of the platform is provided in Figure 7.

Design Variables:

There are 5 sizing design variables as shown in Figure 7. They are the length and width of the beam, and the distances from the centerline to the outer edge of each layer. There are also 3 combinatorial variables that are the materials used for each of the layers. There are 3 potential material selections for each layer of the beam. The relevant properties of each material are provided in Table 1.

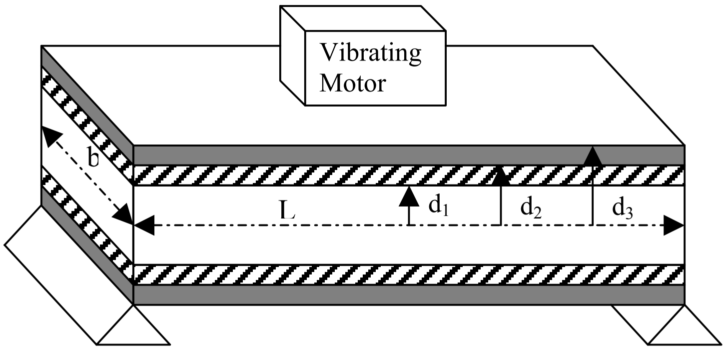

Figure 7: Diagram of Vibrating Platform

Material	Density (ρ) (kg/m^3)	Modulus (E) (N/m^2)	Cost (C) (\$/m^3)
M1	100	1.6 x 10^9	500
M2	2,770	70.0 x 10^9	1,500
M3	7,780	200.0 x 10^9	800

Table 1: Material Properties.

There are side constraints on each of the sizing design variables and they are given in Table 2.

Variable	Lower Bound	Upper Bound
L (m)	3.00	6.00
b (m)	0.35	0.50
d_1 (m)	0.05	0.50
d_2 (m)	0.20	0.50
d_3 (m)	0.20	0.60

Table 2: Side constraints for sizing variables.

Objectives:

The two objectives are to maximize the fundamental frequency (Hz) of the beam and to minimize the cost of material. The formulations are provided in Equations 3 through 6.

Maximize $$f_1 = -\left(\frac{\pi}{2L^2}\right)\left(\frac{EI}{\mu}\right)^{1/2} \quad (3)$$

Minimize $$f_2 = 2b\left[C_{m1}d_1 + C_{m2}(d_2 - d_1) + C_{m3}(d_3 - d_2)\right] \quad (4)$$

Where:

$$EI = \left(\frac{2b}{3}\right)\left[E_{m1}d_1^3 + E_{m2}(d_2^3 - d_1^3) + E_{m3}(d_3^3 - d_2^3)\right] \quad (5)$$

$$\mu = 2b\left[\rho_{m1}d_1 + \rho_{m2}(d_2 - d_1) + \rho_{m3}(d_3 - d_2)\right] \quad (6)$$

ρ, E, and C represent the density, modulus of elasticity, and cost per unit volume respectively for each material as indicated in Table 1.

Constraints:

There are 3 inequality constraints posed in the original problem used to limit:

1. the mass of the beam;
2. the thickness of layer 2; and
3. the thickness of layer 3.

We have added 2 additional inequality constraints to ensure that layers 2 and 3 do not have negative thickness ($d_1 \leq d_2 \leq d_3$) (Note that this formulation allows either of layer 2 or 3 to have zero thickness, as was the intent of the original problem).

Finally, the original problem specification states that the materials used for each of the layers must be mutually exclusive (the same material may not be used for more than one layer at a time) but no concession was made for this in the problem formulation. Therefore, three more constraints are added to enforce this stipulation. This requirement will be forgone for any layer having zero thickness. These three constraints will be posed as not-equality constraints. The term "not-equality" constraints refers to a constraint, as shown in Equations 12-14, that cannot equal a certain value. This type of constraint, being the opposite of an equality constraint, is therefore termed a not-equality constraint.

The formulation for each constraint is provided in Equations 7 through 14. They are given in the same order in which they are discussed previously.

g_1: $$\mu L - 2{,}800 \leq 0 \quad (7)$$

g_2: $$d_2 - d_1 - 0.15 \leq 0 \quad (8)$$

g_3: $$d_3 - d_2 - 0.01 \leq 0 \quad (9)$$

g_4: $$d_1 - d_2 \leq 0 \quad (10)$$

g_5: $$d_2 - d_3 \leq 0 \quad (11)$$

If (d_2 not equal to d_1) z_1: $$M_2 - M_1 \neq 0 \quad (12)$$

If (d_2 not equal to d_1 and d_3 not equal to d_2) z_2: $$M_2 - M_3 \neq 0 \quad (13)$$

If (d_3 not equal to d_2) z_3: $$M_1 - M_3 \neq 0 \quad (14)$$

The final characteristics of the adapted problem are:

- 2 objective functions;
- 8 design variables; and
- 8 constraints (5 inequality, 3 not-equality).

5 Results

The following results were gathered by running multiple separate multi-objective genetic algorithms on the previous problem simultaneously in conjunction with cloud visualization.

5.1 Solution Methodology

Because the layer materials must be mutually exclusive for a design to be feasible, multi-objective genetic algorithms are used to solve the problem. Each was constrained to one of the 6 portions of the design space containing potentially feasible points (those for which the layer materials are mutually exclusive). Each algorithm was run 30 generations at a time and then the user was given a choice of continuing with the optimization, visualizing the current data, discontinuing some MOGAs, or ending the optimization. From within Cloud Visualization, the user is able to view some or all of the individual populations individually. Note that it is possible for the user to view the optimization at any interval. Thirty generations are chosen somewhat arbitrarily. A designer could choose to visualize after each generation and have the display continuously updated.

In the upcoming section, the captures of the performance spaces display the two objectives versus one another and the desirable region is the lower right corner. The captures of the design space will be limited to those containing the 3 layer-thickness sizing variables along the three axes for brevity.

5.2 Solution Process

Figure 8 shows the designs present in all populations after 60 generations. The performance space is on the top and the design space is on the bottom (also in Figures 9-12). Cloud Visualization colors the designs with respect to all other designs being displayed.

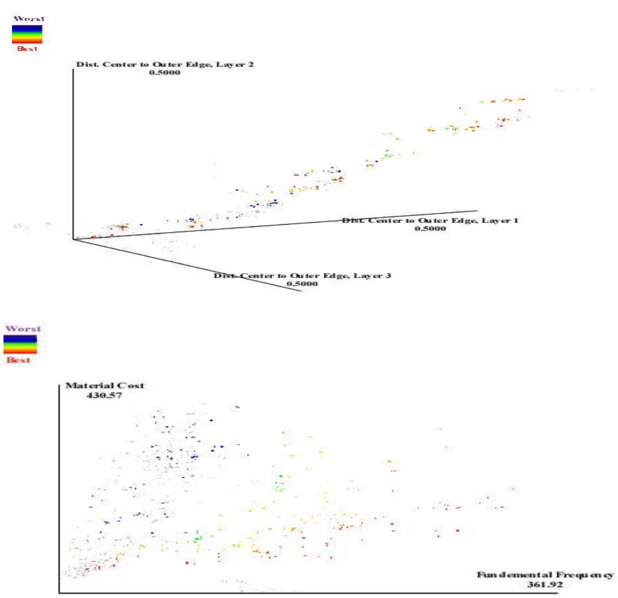

Figure 8: Design data after 60 Generations.

Based on the data shown in the Figure 8, it was decided to discontinue the designs that appear in cyan in Figure 9.

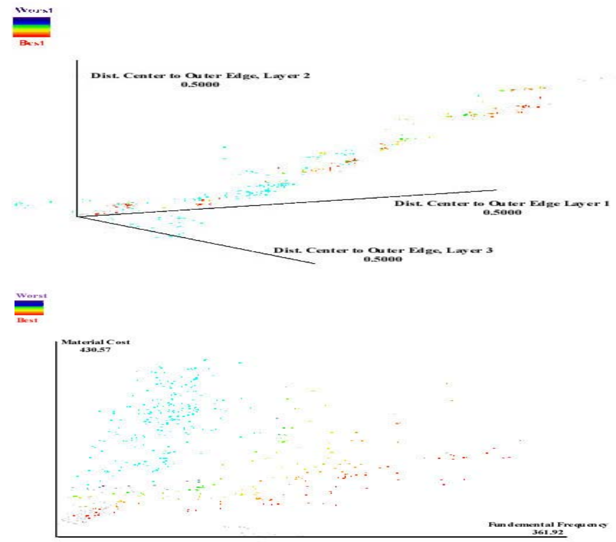

Figure 9: Design data after 60 Generations showing discontinued designs.

Following this, another 60 generations were run with the remaining points. The discontinuation of points in the previous visualization caused the extinction of the fourth MOGA. This means that the resulting population was too small to sustain itself. A population is too small to sustain itself if, according to the crossover rate, no children will be produced. In this case, the remaining members of the extinct MOGA are given an opportunity to join another MOGA's population. The resulting data from the next 60 generations of the remaining MOGAs is shown in Figure 10.

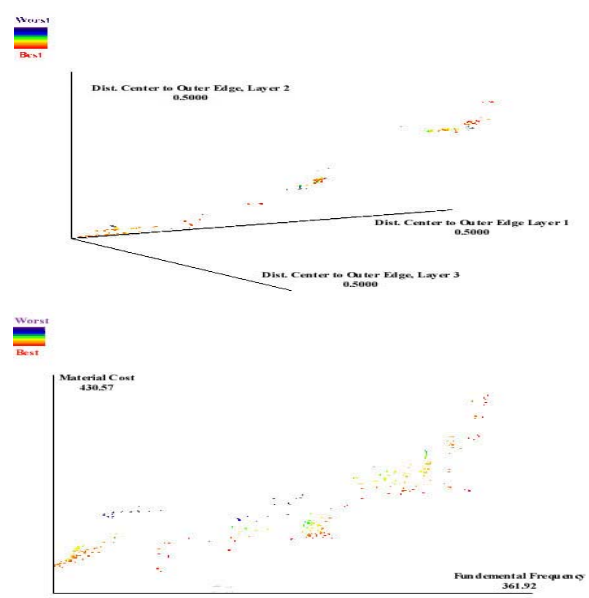

Figure 10: Design data after 120 Generations.

Based on the captures in Figure 10, some additional discontinuations were decided on as shown by the cyan points in Figure 11. These designs were discontinued because in the performance space, they were clearly not close to the efficient frontier and were dominated by other better designs.

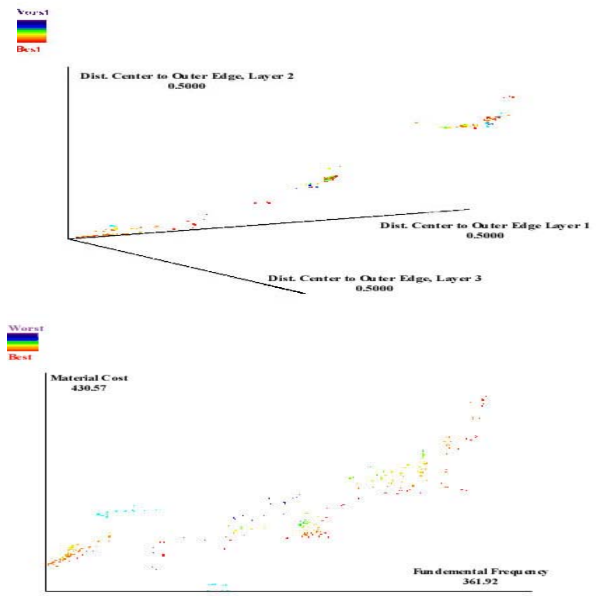

Figure 11: Design data after 60 Generations showing discontinued designs..

Continuing as above, the final Pareto set is shown in Figure 12. Only two MOGAs remain at this point and together they contain the entire non-dominated frontier which may or may not be the true Pareto frontier.

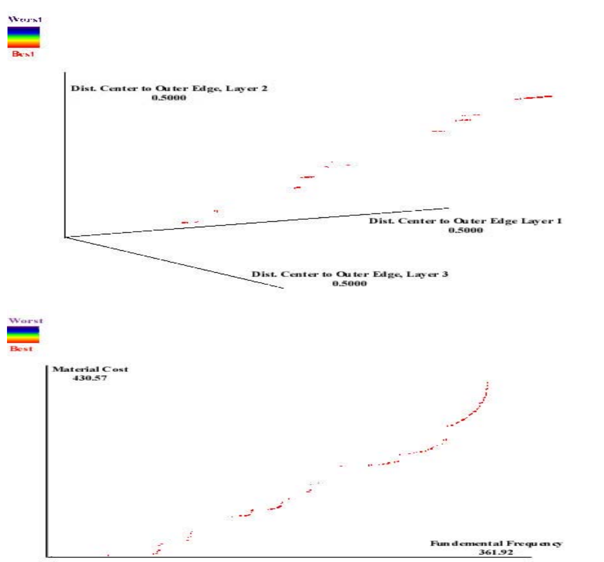

Figure 12: Resulting Pareto Set for Vibrating Platform problem.

For the interested reader, the extremes of the Pareto set are at (396.06Hz, \$216.44) and (99.16Hz, \$75.87). There are 138 total Pareto points and all total, there were 25,210 total function evaluations (approximately 183 evaluations per non-dominated point). Other results using a single MOGA required over 33,000 evaluations to achieve a similar non-dominated frontier.

While the focus of this paper is not on benchmarking the MOGA that was developed (see [24] for a thorough discussion of the MOGA and its effectiveness), the focus was on presenting the visualization tools that are coupled with the MOGA to provide engineers and decision makers with effective information regarding their design system. While it is difficult to benchmark visualization tools, the decision maker will have the final say as to what kind of tool is useful and effective. While providing the decision maker with 138 Pareto points efficiently, he/she must still decide on one of these points to produce/manufacture. This is a non-trivial problem and one that requires assessment of a decision maker(s) preferences and utilities towards risk. While beyond the scope of this paper, current research is investigating this natural evolution of this work.

6 Conclusions

Cloud Visualization posses many attributes that are desirable for a design optimization visualization package for both single and multiple objective problems. It can accommodate large amounts of data since each design appears as a single point on the screen (the size of the point can be adjusted for easy viewing). It also accommodates multidimensional data through the use of cascading windows. Dependent variables can be compared with one another by investigating the performance space. Correlations between independent and dependent variables can be seen by using the two spaces simultaneously. The user interface is provided through pop-up menus and is simple to use for anyone with basic computer skills.

The user is able to interact with the data using the interactive capabilities discussed in Section 3.1. Furthermore, the user can customize the output to his/her preferences by selecting which variables lie along which axes, selecting which type of points are displayed (feasible, infeasible, Pareto, etc.), by specifying which spaces are viewed at a given time, and by choosing the color scheme. The developments of this paper can be applied to both single and multiple objective problems, providing decision makers with an effective interactive presentation of design information in multiple dimensions and across both design and performance spaces simultaneously. This breadth of information has the potential to make decision making before, during, and after optimization processes as well as optimization processes themselves more efficient and effective.

While addressing a set of issues in visualization of single and multiple objective optimization problems, there are still a number of issues under investigation. High dimension problems still pose a challenge in visualization. Novel schemes that capitalize on interactive approaches using virtual reality are possible approaches to the high dimension problems.

Acknowledgements

We acknowledge the support of the National Science Foundation, grants DMII-9875706 and DMII-0115444 in this work.

References

[1] Afimiwala, K.A., and Mayne, R.W., 1979, "Interactive Computer Methods for Design Optimization," *CAD Journal*, **11**, pp. 201-208.

[2] Stuart, R., 2001, *Design of Virtual Environments*, Barricade Books, Fort Lee, NJ.

[3] Mecham, M., 1997, "Aerospace Chases the Software Boom," *Aviation Week & Space Technology*, McGraw Hill, October 6, pp. 46-48.

[4] Stewart, P., and Buttolo, P., 1999, "Putting People Power into Virtual Reality," *Mechanical Engineering Design*, ASME, pp. 18-22.

[5] Furlong, T. J., Vance, J. M., Larochelle, P. M., 1999, "Spherical Mechanism Synthesis in Virtual Reality", *Journal of Mechanical Design*, **121**, pp. 515-520.

[6] Beazley, D.M., and Lomdahl, P.S., 1996, "Lightweight Computational Steering of Very Large Scale Molecular Dynamics Simulations," presented at *Supercomputing*, Pittsburgh, Pennsylvania.

[7] Parker, S., Weinstein, D., and Johnson, C., 1997, **"**The SCIRun Computational Steering Software System," *Modern Software Tools in Scientific Computing*, pp. 1-40.

[8] van Liere, R., Mulder, J.D., van Wijk, J.J., 1996, "Computational Steering," presented at *HPCN Europe*, Brussels.

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

[10] Winer, E.H. and Bloebaum, C.L., 2002, "Development of Visual Design Steering as an Aid in Large Scale Multidisciplinary Design Optimization - Part I and II," *Structural and Multidisciplinary Optimization*, to appear.

[11] Messac, A., and Chen, X., 2000, "Visualizing the Optimization Process in Real-Time Using Physical Programming," *Engineering Optimization Journal*, **32**(5).

[12] Mecham, M., 1997, "Raytheon Integrates Product Development," *Aviation Week & Space Technolog*y, McGraw Hill, October 6, p. 50.

[13] Helig, M. H., 1992, "El Cine del Futuro: The Cinema of the Future," *Presence*, **1**(3), pp. 279-292.

[14] Eddy, W.F. and Mockus, A., 1996, "Dynamic Visualization in Modeling and Optimization of Ill Defined Problems," In *State of the Art in Global Optimization: Computational Methods & Applications*, eds. Floudas, C.A., and Pardalos, P.M., Kluwer Academic Publishers, Boston, pp. 499-520.

[15] Das, I. and Dennis, J. E., 1997, "A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto set Generation in Multicriteria Optimization Problems," *Structural Optimization*, **14**, pp. 63-69.

[16] Messac, A., Sundararaj, J. G., Tappeta R. V., and Renaud, J. E., 2000, "The Ability of Objective Functions to Generate Points on Non-Convex Pareto Frontiers," *AIAA Journal*, **38**, pp. 1084-1091.

[17] Arrow, K. J., 1951, *Individual Values and Social Choice*, Wiley (2nd ed, 1963).

[18] Kasprzak, E., and Lewis, K., 2001, "Pareto Analysis in Multiobjective Optimization Using the Colinearity Theorem and Scaling Method", *Structural and Multidisciplinary Optimization*, **22**(3), pp. 208-218..

[19] Bangay, S., 1998, "Visiview, A System for the Visualization of Multi-Dimensional Data," *IS&T/SPIE Conference on Visual Data Exploration and Analysis V.*, Vol. 3298 - 0277-786X/86/98.

[20] Tufte, E.R., 1997, *Visual Explanations*, Graphics Press, Cheshire, Connecticut.

[21] Tufte, E.R., 2001, *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, Connecticut.

[22] Wu, J. and Azarm, S., 2000, "Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set," *ASME Design Engineering Technical Conferences*, Baltimore, MD, DETC2000/DAC-14233.

[23] Messac, A., 1996, "Physical Programming: Effective Optimization for Computational Design", *AIAA Journal,* **34**(1), pp 149-158.

[24] Eddy, J., 2001, "The Use of Genetic Programming and Visualization to Facilitate Multiobjective Design Optimization", M.S. Thesis, University at Buffalo.