Exploring the Effectiveness of Using Graveyard Data When Generating Design Alternatives

Garrett Foster
Research Assistant
e-mail: gdfoster@ncsu.edu

Scott Ferguson
Assistant Professor
e-mail: scott_ferguson@ncsu.edu

North Carolina State University,
Department of Mechanical and Aerospace
Engineering Raleigh, NC 27695

The objective of this paper is to demonstrate that unique alternative designs can be efficiently found by searching the discarded data (or graveyard) from a multiobjective genetic algorithm (MOGA). Motivation for using graveyard data to generate design alternatives arises from the computational cost associated with real-time design space exploration of multiobjective optimization problems. The effectiveness of this approach is explored by comparing (1) the uniqueness of alternatives found using graveyard data and those generated using an optimization-based search, and (2) how alternative generation near the Pareto frontier is impacted. Two multiobjective case study problems are introduced—a two bar truss and an I-beam design optimization. Results from these studies indicate that using graveyard data allows for the discovery of alternative designs that are at least 70% as unique as alternatives found using an optimization-based alternative identification approach, while saving a significant number of functional evaluations. Additionally, graveyard data are shown to be better suited for alternative generation near the Pareto frontier than standard sampling techniques. Finally, areas of future work are also discussed. [DOI: 10.1115/1.4024913]

Keywords: design alternatives, multiobjective optimization, multiobjective genetic algorithms, design space exploration, MGA

1 Introduction

Whether exploring a multiobjective performance space or a multidimensional tradespace, a designer must make a final selection from an often large number of alternatives. In multiobjective optimization problems, a designer must navigate the tradeoffs between competing objectives [1–3] and select the location on the Pareto frontier that best satisfies the preference structure of the designer. Similarly, tradespace analysis involves executing broad trade studies to understand which designs meet minimum requirements while locating those designs that best meet the goal of the project [4–7]. To aid in this selection process, design approaches have been created that capture the preference structure of the designer using hypothetical alternatives [8,9] or reduce the problem to a single metric using value or utility functions [10–12]. However, a different mindset on this selection process suggests a designer "shop" [13] amongst design alternatives to iteratively update their preference structure.

In this "design by shopping" paradigm, designers often use software tools capable of multidimensional visualization to facilitate design steering [6,14,15]. These environments allow designers to simulate large numbers of alternatives and actively select interesting regions of the space for further exploration. Further, design steering provides an environment where the designer (or the human) stays "in-the-loop" throughout the selection process. This is especially important at the earliest stages of design, as designers often work with system models that have imprecision or uncertainty [16–21]. Additionally, many models cannot, or do not, model all factors that may impact performance or a designer's overall preference structure for a design [22,23].

The presence of model uncertainties and unmodeled factors, and the desire to keep the designer in-the-loop, emphasizes the need for rapidly generated alternatives when selecting a final design. This has been highlighted in recent efforts involving the tradespace exploration of adaptable and resilient systems [24] and by the MDO community [25]. Uncertainties and requirement changes require alternatives to be "more thoroughly explored and kept open longer than they are today" [26]. This paradigm also provides opportunities to further the field of mass customization, allowing customers to select a region of the space and generate customized products that may also be unique.

Searching for the most unique alternative requires formulating an optimization to search the design space. However, for problems with even modest levels of computational burden, this optimization could require significant computational resources. Moreover, if multiple alternatives are desired, the computational cost increases since each alternative must be found sequentially (i.e., to maximize uniqueness, the second alternative needs to consider the first alternative during the search). Combined, this can make alternative identification a prohibitively expensive undertaking.

This work investigates whether the discarded information from a multiobjective genetic algorithm (MOGA) [27–29]—the set of dominated points explored in the process of finding the Pareto frontier—can be "re-used" to help a designer efficiently identify alternatives with similar performance but distinct design space locations. By using designs that have been previously evaluated and discarded, the process of identifying alternatives is dramatically simplified to three straightforward operations: (1) filter out any designs that do not possess performance within the indifference threshold; (2) calculate the uniqueness of the remaining designs; and (3) pull out the design with the maximum uniqueness. This process involves no objective function evaluations, and finding multiple alternatives is not burdensome since the operations used above are essentially "free" to modern computers (they can be performed in a matter of seconds).

As further described in Sec. 3, a designer starts by selecting a point of interest (POI) from the Pareto frontier. An assumption of this work is that the problem's design space was sampled to locate the set of nondominated points, and that dominated designs were saved. Next, a performance indifference threshold is defined for

Contributed by the Computers and Information Division of ASME for publication in the Journal of Computing and Information Science in Engineering. Manuscript received January 11, 2013; final manuscript received June 7, 2013; published online August 19, 2013. Assoc. Editor: Joshua D. Summers.

each objective, providing freedom in the performance space with which to find unique alternatives. Finally, an alternative is identified that maximizes the difference in design space location (compared to the point of interest) while remaining in the performance space indifference threshold. A diverse set of solutions gives the designer a multitude of options to help mitigate unforeseen issues such as a change in requirements instead of forcing them restart from scratch.

While graveyard data reduce cost by preventing re-evaluation of designs, the quality of alternatives found when using a MOGA must be explored. To understand the impact on solution quality, this work investigates how the uniqueness of alternatives found from graveyard data compare to alternatives found using an optimization approach. As recent research has also highlighted the interest in exploring near-Pareto optimal designs [30], the density of alternatives near the Pareto frontier from the graveyard data is compared to datasets created from the more conventional sampling methods (Latin Hypercube Sampling and Exhaustive Search).

In Sec. 2, a brief review of relevant literature is provided. Section 3 describes the alternative identification approach in greater detail. Solution quality in terms of uniqueness is explored in Sec. 4, and the density of alternatives near the Pareto frontier is investigated in Sec. 5. Finally, conclusions are provided and future work is identified in Sec. 6.

2 Background

The significance of having alternative designs present during the design process can be traced to the desire for design freedom [31]. Design freedom has been described as the ability of a system to be "adjusted while still meeting its design requirements" [31]. Two strategies are possible for attaining design freedom: (1) maintain it when possible or (2) create it when needed. Set-based design, for example, is a popular design practice for maintaining design freedom [32–34]. Here, the fundamental idea is to delay restricting the design space until as late as possible. Recent work by Madhavan et al. [35] has shown that in an industrial setting, set-based design approaches reduce the number of iterations between design teams and provide a library of back-up design options. In a similar light, the skewboid method [36] could encourage more design options to be maintained by relaxing the filtering that occurs during dominance checking. Other approaches include the use of physical programming to differentiate solutions based on their quality [37,38], where all designs within the same class are viewed as having an equivalent performance. Other work has explored the use of target sets to decompose the design space and identify optimal solutions [39]. Further opportunities can be found in robust design, where research has characterized the effect of design variable variations on performance [40] while applying regionalized sensitivity analysis [41] and internal-reduction measures [42].

Rather than attempting to maintain design freedom throughout the process, a slightly different approach is to create design freedom when it is needed. This strategy has been significantly enabled by research in multidimensional visualization and the decision sciences. Work in this area has focused on providing software tools that enable visualization [4,43–45], user-guided filtering mechanisms [6,46] and space pruning techniques [47,48], identification of feature rules that lead to interesting areas of a performance space [49], design clustering using neural networks [50], and the creation of metrics to facilitate initial design selection [5,51].

Once an initial point has been selected, alternative designs can then be created for the designer to choose from. Research in this area has explored the generation of alternatives using decision trees [52], grammar rules [53], and-or graphs [54], p-graphs [55], and morphological charts [56]. Alternative generation techniques with a more mathematical foundation have used genetic algorithms [57–60], nearest neighbor calculations [61], and the notion of designer equivalence to identify pockets of unique designs [62]. Previous work by the authors theorized that design freedom could be efficiently created exploring the mapping relationship between the performance and design spaces in multiobjective optimization problems. Initial work on this concept [63] focused on giving designers a set of starting designs rather than an arbitrary performance target. Simov [64] built upon this work and discovered that the larger relative mapping volumes tended to occur in the dominated region of the performance space. The outcome of this study concluded that achieving design freedom at a specific performance level required the selection of a design in the dominated region of the performance space.

Rather than relying on the limited design freedom associated with performance-to-design space mappings, this work builds upon the modeling to generate alternatives (MGA) approach [17–19,65–67]. MGA identifies design alternatives that maximize design space distance from an initial point while satisfying designer-defined performance loss constraints. Developed in the early 1980 s and initially applied to land use planning problems [65], this approach was originally developed for linear programming problems. Recent efforts by the authors have demonstrated the applicability of the approach to multiobjective nonlinear problems [68–70].

Research efforts have explored many variations of the MGA approach [16], and algorithms based on random design variable selection and maximizing the difference between design variables have demonstrated the greatest effectiveness. Both algorithms operate on the same basic principle, which is maximizing the difference between the variables in two different designs while satisfying original problem constraints and user-defined performance thresholds. They differ in the difference calculation. The first algorithm applies a special type of weighted sum to the variables. Essentially, variables that were randomly selected would receive a weight of "1," while the variables that were not randomly selected would receive a weight of "0" causing them to be ignored during the optimization process. The second variation, hop skip jump (HSJ), does not weigh the variables and considers them all equally during the optimization process. Testing both variations showed that they possessed similar performance [65].

Kripakaran and Gupta, for example, used the HSJ variation to optimize a Moment-Resisting Steel Frame [66], where their goal was to identify locations that would minimize the cost of the structure while maintaining the necessary strength to resist wind loading. More recent applications of MGA include the use of genetic algorithms to help solve watershed development problems [67] and an application of MGA to energy modeling [19]. In the watershed development problem, a genetic algorithm was used to start the MGA search process before handing control off to a more efficient algorithm. Meanwhile, the energy modeling work used MGA to explore a conceptual model of the U.S. electric sector.

The goal of this work is to make the MGA process more efficient by reusing data evaluated during the initial MOGA optimization to facilitate real-time alternative generation and design space exploration. This is done by keeping a "graveyard" of previously evaluated designs; a capability enabled by the near limitless memory on modern computer. By comparing each incoming design with the graveyard, the number of redundant objective function evaluations performed during an optimization can be reduced [71]. Rather, the computational cost is associated with mining a database of previously evaluated points. Such computational savings are analogous to metamodeling [72–74], where existing data are used to estimate the performance values of new designs. By reducing the need to evaluate the objective function of each new possible design, optimizations using metamodels experience dramatically reduced computational overhead with only minor amounts of inaccuracy [74].

This graveyard enhanced MGA (GE-MGA) approach, as well as the traditional MGA approach, is presented in greater detail in Sec. 3. These two approaches are then applied in Sec. 4 to explore their effectiveness.

3 Alternative Identification Approach

As mentioned in Sec. 2, MGA is an approach that identifies alternative designs that maximize design space distance from an initial point while satisfying performance loss constraints. In this research the HSJ variation of MGA is used due to its simpler formulation, as shown in the following equation:

$$\begin{aligned} \text{Max: } d_{ab} &= \sum_{i=1}^{n} |a_i - b_i| \\ \text{S.T.: } F_j^b &\leq F_j^a + \delta F_j \quad j = 1\ldots\text{m} \\ G_k^b &\leq C_k \qquad k = 1\ldots p \end{aligned} \tag{1}$$

In this formulation, the goal is to maximize the distance (d_{ab}) from the original design vector (a) to the potential alternative design vector (b). The design variables, a_i and b_i, should be limited to either quantity or measurement values otherwise the distance calculation will not correspond to design uniqueness. Distance is measured using a one-norm distance calculation; however, two-norm distance could also be used; it is just more sensitive to the scaling applied to the design string.

To ensure the alternatives identified exist within the desired region of the performance space, the performance of an alternative's objective function values (F_j^b) must be less than the original design's objective function values (F_j^a) plus a small indifference threshold (δF_j), assuming minimization.

Finally, each of the constraint values of the potential alternative (G_k^b) must be less than constraint values (C_k) of the original problem, again assuming minimization. Here, n refers to the number of design variables, j refers to the number of objectives present in the problem, and k represents the number of constraints.

The process used to find alternative designs contains four steps, each of which is explained in Sec's 3.1 – 3.4:

(1) Select a point of interest
(2) Specify indifference threshold(s)
(3) Formulate distance measure
(4) Find alternative(s)

As previously mentioned, the first step aligns itself with the "design by shopping" paradigm associated with exploring the results of multiobjective optimization problems and multidimensional tradespaces. Steps 2 and 3 are used by the decision maker to specify the set of parameters needed to constrain the generation of alternative designs. Finally, the fourth step uses the formulation in Eq. (1) to identify unique alternative designs that possess similar performance. However, the application of this formulation is the difference between the proposed GE-MGA approach and the traditional MGA approach.

3.1 Selecting a Point of Interest. To investigate the effectiveness of the GE-MGA approach, two multiobjective case study problems are explored in Sec. 4. To enable comparison, both the graveyard enhanced and traditional versions of the MGA process will be applied. For both versions, the first step is to choose a point of interest from the two-objective performance space. In this research, a point of interest is chosen by applying a weighted L1 norm to a normalized, MOGA generated, Pareto frontier. The performance space is normalized according to the extreme points on the Pareto frontier. To explore different regions of the Pareto frontier, three sets of weights—[0.9, 0.1], [0.5, 0.5], and [0.1, 0.9]—are used.

3.2 Specification of Indifference Threshold(s). Step 2 of the approach specifies that an indifference threshold is chosen by the designer. Typical starting values for the indifference threshold start around 10–15% of the point of interest's objective function values [23]. However, when these percentage values are applied to the point of interest's performance value, the resulting performance space indifference threshold is nonuniform across different points of interest. To overcome this, and allow comparison across points of interest, the percentage values are applied to the range of objective values captured by the Pareto frontier instead. In this research, alternatives are found using percentages of 5%, 10%, 20%, and 50%.

3.3 Formulation of Distance Measures. Step 3 of the approach requires that the designer formulate a distance metric. As seen in Eq. (1), one-norm distance is chosen for this research; however, any number of distance metrics could be formulated and used. Previous research by the authors found the one-norm distance metric to be fairly robust for this application [69]. To improve on solution robustness, the design strings are also normalized according to the variable bounds provided in each case study problem, as shown in Eq. (2). In this equation, the ith design variable (x_i) is normalized by first subtracting the lower bound of the variable (lb_i). Next, that value is divided by the range of the bounds, which is equal to the upper bound (ub_i) minus the lower bound (lb_i). Finally, the resulting normalized variable (x_i^{norm}) will have a value on the range between 0 and 1 inclusive

$$x_i^{\text{norm}} = \frac{x_i - lb_i}{ub_i - lb_i} \tag{2}$$

If multiple alternatives are desired, a slight variation to the distance metric described in Eq. (1) is necessary. This is because the approach must take into consideration alternatives that have already been found; otherwise, the same design would be returned multiple times. To prevent this, one of two approaches can be used. Both start by calculating the distances from the potential alternative to both the point of interest and the already discovered alternative(s). The first approach simply adds up all these distances up and returns the result. Meanwhile, the second approach returns the minimum of these distances. The advantage of the second approach is that it will not return an already found alternative. However, the first approach is useful only if extreme or corner points are desired. In this work, multiple alternative designs are not found since the identification of additional alternatives is dependent on the location of the first alternative. To eliminate any potential side effects of this dependency from the results, only the first alternative is identified.

3.4 Identification of Alternatives. Once the parameters from the first three steps are in place, alternatives can be found. This fourth step is where the GE-MGA approach differs from the traditional MGA approach. In the traditional MGA approach, an optimization algorithm is used to find the alternatives. Since the formulation of the distance measure in Eq. (1) is single objective, a number of optimization algorithms can be used. In this work, the Matlab pattern search function is used [75] due to the ease of implementation, robustness, and repeatability of the result. The pattern search algorithm [76,77] is a type of direct search algorithm, meaning it does require gradient information to optimize the problem. This does make pattern search a less efficient algorithm; however, this was a tradeoff the authors were willing to make as other more efficient algorithms found alternatives that were less unique. This is further discussed in Sec. 4.

To perform the GE-MGA approach, the MOGA first needs to store the variable values and associated objective function values of all the designs it evaluates. This was implemented in the Matlab MOGA by creating an output function that stored the population and scores from the state variable [78]. Further, since the Matlab MOGA does not provide a convenient way to check the status of constraints, an additional objective was added to each case study that tracked the number of constraint violations. After running the MOGA, a dataset would be saved containing the necessary information for all evaluated designs. Finally, finding the most unique alternative for a chosen point of interest from this dataset followed the following process:

(1) Filter out any designs that violated a constraint
(2) Filter out any designs that fall outside the indifference threshold
(3) Calculate the uniqueness of the remaining designs using the distance metric specified
(4) Choose the design with the largest uniqueness value

In the following two Sections 5 & 6, the graveyard enhanced and traditional MGA approaches are compared using two multiobjective case study problems. These case study problems are introduced in the next Sec. 4. Additionally, the uniqueness of alternatives found using graveyard data and those generated using an optimization formulation is investigated. The impact of each approach on alternative generation near the Pareto frontier is explored is Sec. 5.

4 Case Study Problem Introductions and Exploration of Alternative Uniqueness

This section explores how the uniqueness of alternatives found from a pre-evaluated dataset compare to alternatives found using an optimization approach. To investigate this, two multiobjective case study problems are used to ensure any identified trends are not problem specific. These problems are introduced in Sec. 4.

4.1 Introduction of Case Study Problems. The first case study problem is a two bar truss optimization problem. Problem formulation, which is provided by Azarm [79], consists of three design variables, two constraints, and two-objective functions. The three design variables are the cross-sectional area of link AC (x_1), the cross-sectional area of link BC (x_2), and the vertical position of the load (y). A diagram of the two bar truss showing these design variables is shown in Fig. 1. The first objective (f_1) in Eq. (3) is to minimize the amount of material used, while the second objective (f_2) in Eq. (4) is to minimize the stress in link AC. Equation (5) shows the stress in link BC being used as the lone constraint (g_1)

$$\text{Min: } f_1 = \frac{20\sqrt{4^2+y^2}}{x_1 y} \leq 0.1\ \text{m}^3 \tag{3}$$

$$\text{Min: } f_2 = x_1\sqrt{4^2+y^2} + x_2\sqrt{1^2+y^2} \leq 100{,}000\ \text{kPa} \tag{4}$$

$$\text{S.T.} g_1 = \frac{80\sqrt{1^2+y^2}}{x_2 y} \tag{5}$$

$$\text{Where: } x_1 > 0\ \text{m}, \quad x_2 > 0\ \text{m}, \quad \text{and} \quad 1 \leq y \leq 3\ \text{m} \tag{6}$$

An I-beam optimization serves as the second case study problem. The formulation of this problem, which is adapted from Hacker [80], has four design variables and two-objective functions. The four design variables (x_1 *to* x_4) can be seen in the diagram shown in Fig. 2. The first objective (f_1) of this case study is to minimize the cross-sectional area as seen in Eq. (7), while the

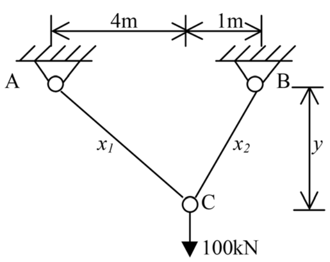

Fig. 1 Diagram of two bar truss

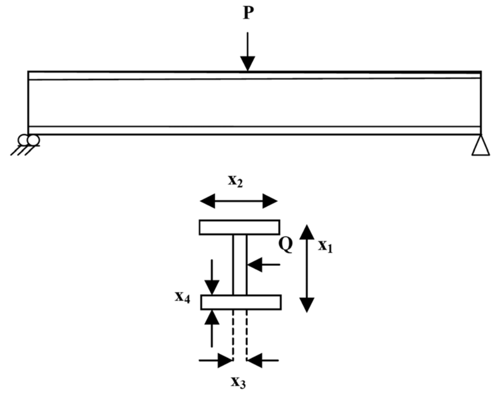

Fig. 2 Cross section of I-beam

second objective (f_2) is to minimize the vertical deflection of the beam subject to loads P and Q as seen in Eq. (8). Equation (9) shows the stress in the beam acting as the lone constraint (g_1)

$$\text{Min: } f_1 = 2x_2x_4 + x_3(x_1 - 2x_4)^3\text{cm}^2 \tag{7}$$

$$\text{Min: } f_2 = \frac{5000}{\frac{x_3}{12}(x_1-2x_4)^3 + \frac{x_2x_4^3}{6} + 2x_2x_4\left(\frac{x_1-x_4}{2}\right)^2}\text{cm} \tag{8}$$

$$\text{S.T.} g_1 = \frac{180{,}000x_1}{x_3(x_1-2x_4)^3 + 2x_2x_4\left(4x_4^2 + 3x_1(x_1-2x_4)\right)} + \frac{15{,}000x_2}{(x_1-2x_4)x_3^3 + 2x_4x_2^3} \leq 16\frac{kN}{\text{cm}^2} \tag{9}$$

$$\text{Where: } 10 \leq x_1 \leq 80\ \text{cm}, 10 \leq x_2 \leq 50\ \text{cm, and } 0.9 \leq x_3, x_4 \leq 5\ \text{m} \tag{10}$$

4.2 Generation of Alternatives Using Traditional MGA and GE-MGA. To investigate the issue of alternative uniqueness, alternative designs were found using both the graveyard enhanced and traditional MGA approaches. To ensure repeatability, the following procedure was followed for each case study problem.

(1) Run 100 MOGAs to convergence using the default settings and store the necessary data
(2) For each MOGA, filter out designs that are either not feasible or not unique
(3) From the filtered dataset find the Pareto frontier
(4) From the Pareto frontier find the three points of interest
(5) For each point of interest, calculate the four indifference thresholds
(6) For each indifference threshold, find a single alternative using both MGA approaches
(7) Store the distance to each alternative and the number of functional calls needed by each approach

Once the data were generated, the percent change for each alternative pairing was calculated using Eq. (11). Percent change allows the results from different problems to be better compared since the distance measures can vary greatly. The hypothesis is that alternatives found by the GE-MGA approach will be fairly close to the alternatives found using the traditional approach (i.e., have low percent change). A graphical representation of the percent change calculation in the design space is shown in Figs. 3(*a*) and 3(*b*). Figure 3(*a*) depicts a scenario where the GE-MGA alternative is less unique than the alternative found using the optimization approach associated with traditional MGA. Here, less unique refers to the fact that the one-norm distance from the POI for the GE-MGA alternative is less than the one-norm distance for the traditional MGA alternative. Conversely, a greater than 0%

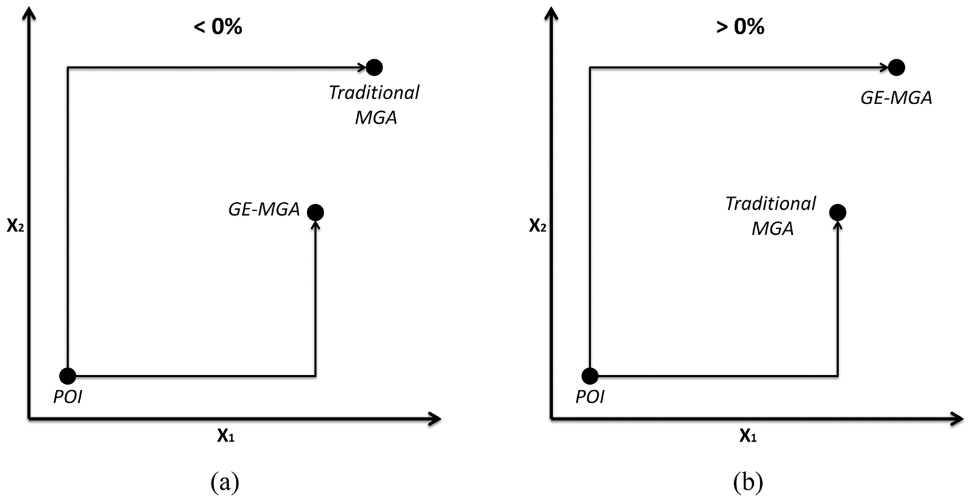

Fig. 3 Illustration of percent change calculation

percent change means that the one-norm distance of the GE-MGA alternative from the POI is greater than the one-norm distance calculated for the traditional MGA alternative.

$$\text{Percent Change} = 100 \times \frac{\text{GE-MGA distance} - \text{Traditional distance}}{\text{Traditional distance}} \quad (11)$$

A histogram count showing the number of alternatives that fell within seven different percent change ranges can be seen in Table 1 for the two bar truss problem and Table 2 for the I-Beam problem.

The values in Tables 1 and 2 indicate that the data are not normally distributed, meaning hypothesis testing, such as a t-test, would be difficult. For both case studies, values less than −50% were found. This is an indication of the indifference threshold filtering out most or all of the potential GE-MGA alternatives. By looking down the (−Inf,−50) column, it is noticed that the number of occurrences with less than −50% change decreases as the indifference threshold increases. Looking at the row of sums at the bottom of each table, it is noticed that the two bar truss results are heavily skewed to the right with over 43% of the alternatives found by GE-MGA having a distance greater than or equal the traditional MGA alternative. The I-beam results also show a large percentage of GE-MGA designs being more unique; however, its largest sum occurred on the [−20,−10) interval with almost 31% of the data points. For both case studies, over 91% of the data points fell within the [−30,Inf) interval. This indicates that most alternatives found by GE-MGA should have a distance measure that is at least 70% as unique as alternatives found using a traditional MGA approach. Finally, for both problems, a large number of data points (over 28%) exist on the [0,Inf) interval. This indicates that GE-MGA found alternatives that were either as unique as, or more unique, than the alternatives found by the traditional MGA approach.

This outcome indicates that the optimization algorithm associated with the traditional MGA approach often became stuck at a local optimum. As mentioned briefly in Sec. 3, the pattern search algorithm was used since it had the lowest number of data points in the [0,Inf) range out of all the available Matlab optimization algorithms (fminsearch [81], ga [82], and fmincon [83]). This indicates that it was better at converging to the global optimum for the two case study problems. Even with the best performing optimization algorithm used as competition, the GE-MGA approach was able to find alternatives that were more unique. However, to provide a "worst case" scenario, one where all the alternatives found were at least as unique as the GE-MGA approach, a hybrid MGA approach was created, and is discussed in Sec. 4.3.

4.3 Generation of Alternatives Using a Hybrid MGA Approach. The hybrid MGA approach uses pattern search to find the final location of the alternative design after the

Table 1 Two bar truss—traditional versus GE-MGA—percent change histogram count

		Percent change ranges						
Point of interest	Threshold (%)	(−Inf,−50)	[−50,−40)	[−40,−30)	[−30,−20)	[−20,−10)	[−10,0)	[0,Inf)
#1	5	21	11	9	9	7	9	34
	10	5	6	9	10	16	10	44
	20	2	0	2	10	25	15	46
	50	0	0	0	4	31	22	43
#2	5	10	9	6	12	9	12	42
	10	2	2	5	11	14	22	44
	20	0	1	2	5	15	36	41
	50	0	0	0	1	23	34	42
#3	5	0	0	3	16	30	28	23
	10	0	0	0	2	17	34	47
	20	0	0	1	17	21	23	38
	50	0	0	0	1	2	21	76
Sum		40	29	37	98	210	266	520

Table 2 I-beam—traditional versus GE-MGA—percent change histogram count

		Percent change ranges						
Point of interest	Threshold (%)	(−Inf,−50)	[−50,−40)	[−40,−30)	[−30,−20)	[−20,−10)	[−10,0)	[0,Inf)
#1	5	9	5	8	12	8	7	51
	10	0	2	6	9	10	10	63
	20	0	0	7	21	16	13	43
	50	0	0	0	13	64	21	2
#2	5	3	3	3	12	12	8	59
	10	0	3	6	26	16	14	35
	20	0	0	8	37	26	13	16
	50	0	0	1	15	49	33	2
#3	5	1	2	9	26	44	9	9
	10	0	1	5	24	52	16	2
	20	0	1	9	26	53	8	3
	50	0	0	1	7	21	16	55
Sum		13	17	63	228	371	168	340

Table 3 Two bar truss—hybrid versus GE-MGA—percent change histogram count

		Percent change ranges						
Point of interest	Threshold (%)	(−Inf,−50)	[−50,−40)	[−40,−30)	[−30,−20)	[−20,−10)	[−10,0)	[0,Inf)
#1	5	16	13	14	12	24	21	0
	10	5	7	7	14	21	46	0
	20	1	1	5	12	34	47	0
	50	0	0	1	6	43	50	0
#2	5	10	12	17	25	23	13	0
	10	2	3	7	12	31	45	0
	20	0	1	3	7	37	52	0
	50	0	0	2	4	40	54	0
#3	5	0	0	3	17	38	42	0
	10	0	0	0	22	44	34	0
	20	0	0	5	43	43	9	0
	50	0	0	23	36	29	12	0
Sum		34	37	87	210	407	425	0

Table 4 I-beam—hybrid versus GE-MGA—percent change histogram count

		Percent change ranges						
Point of interest	Threshold (%)	(−Inf,−50)	[−50,−40)	[−40,−30)	[−30,−20)	[−20,−10)	[−10,0)	[0,Inf)
#1	5	6	11	23	33	20	7	0
	10	0	2	17	39	35	7	0
	20	0	0	11	40	40	9	0
	50	0	0	1	11	67	21	0
#2	5	3	7	9	30	36	15	0
	10	0	1	13	32	44	10	0
	20	0	0	8	45	40	7	0
	50	0	0	0	21	66	13	0
#3	5	0	3	7	22	45	23	0
	10	0	1	3	13	54	29	0
	20	0	0	4	12	56	28	0
	50	0	0	0	9	55	36	0
Sum		9	25	96	307	558	205	0

GE-MGA result is used as a starting point. To maximize comparability with the previous results, the same points of interest from the previous 100 MOGAs were used. After finding the new alternatives, new percent change values were calculated using the hybrid alternative's distance in place of the traditional alternative's distance. Table 3 shows the histogram count for the two bar truss problem and Table 4 shows the histogram count for the I-beam problem.

Table 5 Two bar truss—alternatives search computational cost

Point of interest	Threshold (%)	GE-MGA	Average number of functional calls: Traditional MGA CI Lower	Traditional MGA CI Upper	Hybrid MGA CI Lower	Hybrid MGA CI Upper
#1	5	0	279	290	239	272
	10	0	280	293	216	237
	20	0	285	297	207	227
	50	0	314	344	206	220
#2	5	0	314	349	383	448
	10	0	296	316	229	270
	20	0	293	312	233	248
	50	0	307	330	235	251
#3	5	0	298	314	246	266
	10	0	296	313	267	281
	20	0	312	328	294	308
	50	0	295	312	326	346
Overall		0	304	310	264	274

Table 6 I-beam—alternatives search computational cost

Point of interest	Threshold (%)	GE-MGA	Average number of functional calls: Traditional MGA CI Lower	Traditional MGA CI Upper	Hybrid MGA CI Lower	Hybrid MGA CI Upper
#1	5	0	515	599	637	769
	10	0	490	542	515	579
	20	0	595	704	565	609
	50	0	469	500	413	452
#2	5	0	677	814	563	630
	10	0	660	765	557	597
	20	0	535	606	454	519
	50	0	401	426	355	382
#3	5	0	531	587	465	514
	10	0	430	464	349	378
	20	0	426	462	364	394
	50	0	354	382	351	372
Overall		0	527	551	481	501

By looking at the [0,Inf) columns in both tables, it is immediately noticeable that all the alternatives found using GE-MGA are less unique than the alternatives found using the hybrid MGA approach as expected. However, over 86% of data points are still within the [−30,Inf) range. The data in Table 3 show that the results of two bar truss problem remain skewed to the right with over 35% of the data being in the [−10,0) range. The distribution of the data in the I-beam problem also stayed roughly the same with the [−20,−10) interval remaining the most heavily populated with over 46% of the data.

Overall, the results from both studies (traditional and hybrid) suggest that using the GE-MGA approach should identify alternatives that are at least 70% as unique as alternatives found using an optimization-based MGA approach. However, recall that the motivation for using GE-MGA is to reduce computational cost. To get a better understanding of the cost savings, 95% confidence intervals on the average number of functional calls used by both the traditional and the hybrid MGA approaches are presented. Note that function calls are used instead of computer time as the number of function calls is not affected by the computer used. Naturally, more function calls correspond with more computational time. Table 5 contains the data for the two bar truss problem, while Table 6 contains the data for the I-beam problem.

For the two bar truss problem, the average number of objective function calls used by the traditional MGA approach was 307 ± 3 while the average number of functional calls used by the hybrid GE-MGA approach was 269 ± 5. The results from 100 independent runs indicate that using the GE-MGA result as the starting point had a statistically significant effect on lowering the number of functional evaluations needed. This finding is repeated in I-beam problem as well. This reduction was roughly a 10% savings in both problems. Nevertheless, the hybrid approach still needed a nontrivial amount of functional calls even with these savings; approximately 270 for the two bar truss problem and approximately 490 for the I-beam problem. The difference in these values could be caused by the different number of design variables.

For the case study problems used, the entire GE-MGA process takes less than a minute once the MOGA finishes. This means that the hybrid MGA approach on the two bar truss problem (the smallest average number of functional calls) would need a functional evaluation time less than 0.25 s to be competitive. However, with even moderately complex systems, a function evaluation can take several minutes, which leads to several hours (possibly even days) of computational savings. This, combined with the results from the previous study, means a designer can now choose to

Table 7 Two bar truss—alternative density—confidence intervals

		Exhaustive search	Latin hypercube CI		MOGA CI	
			Lower	Upper	Lower	Upper
Feasible		392	422	426	598	680
Indifference thresholds	50%	79	362	373	501	564
	20%	0	278	296	364	407
	10%	0	188	209	261	292
	5%	0	102	119	169	192

Table 8 I-beam—alternative density—confidence intervals

		Exhaustive search	Latin hypercube		MOGA	
			Lower	Upper	Lower	Upper
Feasible		748	1135	1139	878	909
Indifference thresholds	50%	129	818	856	544	566
	20%	0	171	190	171	183
	10%	0	38	43	53	59
	5%	0	9	11	15	18

tradeoff alternative uniqueness (less than 30%) for several hours of computational savings. With this in mind, Sec. 5 explores whether the MOGA generated dataset is better suited for alternative generation than a dataset created from a more common sampling technique near the Pareto frontier.

5 Comparing the Density of Generated Alternatives

The investigation in Sec. 4 revealed that alternative generation using previously discarded (i.e., graveyard) data from a MOGA offers many opportunities for computational saving. Further, this approach occasionally led to the discovery of alternatives with greater uniqueness than alternatives found using a more traditional optimization-based search approach. With these results in mind, this section investigates if the MOGA generated dataset used in Sec. 4.2 is better suited for alternative generation near the Pareto frontier than a dataset generated from a more conventional sampling technique. The hypothesis is that a MOGA generated dataset will maintain a larger set of possible alternatives as the indifference threshold shrinks; and by providing this larger set of options, the MOGA generated dataset should therefore perform better.

To investigate this hypothesis, the density of alternatives created using two conventional sampling techniques will be compared to the density of alternatives created using a MOGA for both case study problems introduced in Sec. 4. The two sampling techniques chosen are Latin Hypercube Sampling [84] and Exhaustive Search [85]. To ensure comparability between datasets, all three sampling techniques are constrained to the same number of unique functional evaluations; 1500 for the two bar truss problem and 2000 for the I-beam problem. These values were chosen based on the average number of unique functional evaluations needed to converge using the default Matlab MOGA settings [78]. To ensure repeatability, 100 independent datasets for each technique were created.

To calculate the alternative density, these datasets were progressively filtered and the number of remaining designs was counted. Since each dataset started with the same number of unique designs, the first filter removed any designs that were infeasible. Next, the datasets were filtered at each of the four indifference threshold percentages, starting at 50% and ending at 5%. For a design to be filtered out by an indifference threshold percentage it needed to have a performance value that was not within the specified indifference threshold of any of the Pareto frontier points.

Examining the results for each sampling technique indicated that the Latin Hypercube and MOGA were found to have data that were nearly normally distributed. Therefore, 95% confidence intervals were computed for both datasets at each filtering stage. Meanwhile, the Exhaustive Search is not stochastic, meaning its alternative density can be represented by a single value at each filtering stage. The number of potential alternatives at each filtering stage can be seen in Table 7 for the two bar truss problem and Table 8 for the I-beam problem.

The results from both Tables 7 and 8 indicate that the MOGA generated designs were denser near the frontier than either the Latin Hypercube or the Exhaustive Search. At 5% and 10% indifference thresholds, the MOGA held a statistically significant lead over the more traditional approaches. Since a large percentage of the unique designs were lost due to infeasibility, this lead would likely stretch to the 20% threshold if the MOGA handled constraints instead of treating them as objectives. Looking at Table 8, the Latin Hypercube search actually starts with more feasible designs. However, its density falls off faster than the MOGA. Not surprisingly, the Exhaustive Search proved to be quite inefficient as it was unable to identify even a single potential alternative within a 20% indifference threshold for either case study problem. Finally, the number of potential alternatives near the Pareto frontier was generally greater in the two bar truss problem than in the I-beam problem. This is likely a cause of the two bar truss problem having a relatively large amount of design freedom present near a portion of the Pareto frontier.

From these findings, it is apparent that the MOGA generated data set (or graveyard) is better suited for alternative generation near the Pareto frontier than a dataset generated from a common sampling technique. Further, it was shown that the MOGA generated dataset maintained the largest set of possible alternatives at the smallest indifference threshold percentages. In Sec. 6, these results, along with the results from Sec. 4, will be summarized and discussed.

6 Conclusions and Future Work

This paper proposes that unique design alternatives can be efficiently located by searching the discarded data from a MOGA (the graveyard), thereby reducing the computational bottleneck associated with alternative generation for real-time exploration of multiobjective problems and multidimensional tradespaces. Two issues explored in this work involve: (1) the uniqueness of alternatives found from a graveyard compared to alternatives found using an optimization approach and (2) if the MOGA-generated graveyard is better suited for alternative generation near the Pareto frontier than a dataset generated from a more conventional sampling technique.

6.1 Discussion of Results. Answering the question of alternative uniqueness was completed in Sec. 4 using the traditional optimization-based MGA approach and the proposed, dataset based, GE-MGA approach. The results demonstrated that the GE-MGA approach identified alternatives that were at least 70% as unique as alternatives found using an optimization-based MGA approach. Additionally, over 28% of the alternatives found using GE-MGA were either as unique, or more unique than, the alternatives found by the traditional MGA approach. This is due to the underlying optimization algorithm within the traditional MGA converging to a local optimum.

A "worst case" scenario was explored by introducing a hybrid MGA approach in Sec. 4.3. The difference between the hybrid MGA and traditional MGA approach was that the hybrid approach used the best alternative identified by the GE-MGA as a starting point. Doing this ensured the optimization algorithm would find an alternative that was at least as unique as the GE-MGA approach. The overall results were largely the same; with the largest difference in the results being that all the alternatives found by GE-MGA were now less than the alternatives found using the hybrid MGA approach.

After confirming that the GE-MGA approach still finds fairly unique alternatives, the computational savings were calculated. Two significant findings came from those results. First, using the GE-MGA result as the starting point had a statistically significant effect on lowering the number of functional evaluations needed; roughly 10% fewer in the 100 independent runs performed. Second, the more efficient hybrid approach still needed a nontrivial amount of functional calls; 269 for the two bar truss problem and 491 for the I-beam problem, on average. For a complex system, a function evaluation can take several minutes (if not hours), meaning to several hours (possibly even days) of computational savings are possible.

To explore the issue of alternative density near the Pareto frontier, MOGA-generated graveyard datasets were compared to datasets created from the more conventional sampling methods (Latin Hypercube Sampling and Exhaustive Search). 95% Confidence intervals were then generated for the two stochastic methods (MOGA and Latin Hypercube). The results indicate that the Exhaustive Search is the worst sampling technique to use, as it evaluated no potential alternatives within 20% of the Pareto frontier. Meanwhile, the MOGA and Latin Hypercube both found potential alternatives at the closet threshold to the Pareto frontier (5%). Comparing the confidence intervals indicated that the MOGA-generated datasets were better suited for alternative generation near the Pareto frontier than the Latin Hypercube datasets as the MOGA had more potential alternatives at both the 5% and 10% thresholds. The MOGA would have likely been superior at the 20% threshold as well; however, the stock Matlab MOGA was inefficient at handling constraints as over half the design evaluated were infeasible.

6.2 Future Work. One area of future work is tuning the MOGA. An initial study by the authors indicates that tuning items such as the selection operator, population size, constraint handling could lead to a 300% increase in the number of potential alternatives found by the MOGA at the smallest indifference threshold of 5%. This kind of improvement would make the alternatives found even more unique; further, reducing the lone advantage, the traditional MGA approach still has. Additionally, it is theorized that a growing population will increase the density of potential alternatives even further.

A second area of future work could improve the uniqueness of the designs by leveraging metamodeling. Recall that using the GE-MGA approach to find a starting point for the hybrid MGA approach yielded greater uniqueness than the GE-MGA approach alone. The downside of using this combination comes from the additional functional calls needed by the hybrid MGA approach. However, if these functional calls could be made on a metamodel the cost would be of little computational impact. Moreover, with the MOGA already generating a dense population of designs near the Pareto frontier, it is hypothesized that an accurate metamodel could be constructed without additional functional calls. This study should be performed on larger more open ended case studies with more than two objectives.

In closing, this research shows that a MOGA-generated dataset can be used to find alternative designs that are at least 70% as unique as alternatives found using an optimization-based MGA approach. By making this tradeoff in uniqueness, a designer is able to save several hundreds of functional calls, which is equivalent to several hours of computation time on a moderately complex model. Lastly, should the alternative(s) found using GE-MGA approach fail to be unique enough for the decision maker, that alternative still serves as a better starting point for the traditional MGA approach; one that leads to 10% fewer functional evaluations as well as increased likelihood of finding the global optimum (i.e., most unique alternative).

Acknowledgment

We gratefully acknowledge support from the National Science Foundation through NSF Grant No. CMMI-0969961 and EEC-1037677. Any opinions, findings, and conclusions presented in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Otto, K. N., and Antonsson, E. K., 1991, "Trade-Off Strategies in Engineering Design," Res. Eng. Des., **3**(2), pp. 87–103.
[2] Mattson, C. A., and Messac, A., 2005, "Pareto Frontier Based Concept Selection Under Uncertainty, With Visualization," Optim. Eng., Kluwer Publishers—Special Issue on Multidisciplinary Design Optimization, Invited Paper, **6**(1), pp. 85–115.
[3] Gurnani, A., Ferguson, S., Donndelinger, J., and Lewis, K., 2005, "A Constraint-Based Approach to Feasibility Assessment in Conceptual Design," Artif. Intell. Eng. Des., Anal. Manuf., Special Issue on constraints and Design, **20**(4), pp. 351–367.
[4] Stump, G., Yukish, M., and Simpson, T. W., 2004, "The ARL Trade Space Visualizer: An Engineering Decision-Making Tool," Proceedings of the Tenth AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, Paper No. AIAA-2004-4568.
[5] Ross, A. M., Hastings, D. E., Warmkessel, J. M., and Diller, N. P. 2004, "Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design," J. Spacecr. Rockets, **41**(1), pp. 20–28.
[6] Stump, G., Lego, S., Yukish, M., Simpson, T. W., and Donndelinger, J. A., 2009, "Visual Steering Commands for Trade Space Exploration: User-Guided Sampling With Example," J. Comput. Inf. Sci. Eng., **9**(4), p. 044501.
[7] Daskilewicz, M. J., and German, B. J., 2012, "Rave: A Computational Framework to Facilitate Research in Design Decision Support," J. Comput. Inf. Sci. Eng., **12**(2), p. 021005.
[8] See, T. K., Gurnani, A., and Lewis, K., 2004, "Multi-Attribute Decision Making Using Hypothetical Equivalents and Inequivalents," J. Mech. Des., **126**(6), pp. 950–959.
[9] See, T. K., and Lewis, K., 2006, "A Formal Approach to Handling Conflicts in Multiattribute Group Decision Making," J. Mech. Des., **128**(4), pp. 678–689.
[10] Hollingsworth, P. M., 2004, "Requirements Controlled Design: A Method for Discovery of Discontinuous System Boundaries in the Requirements Hyperspace," Ph.D. thesis, Georgia Tech, Atlanta, GA.
[11] Ross, A. M., and Hastings, D. E., 2006, "Assessing Changeability in Aerospace Systems Architecting and Design Using Dynamic Multi-Attribute Tradespace Exploration," AIAA Space 2006, San Jose, CA, Paper No. AIAA-2006-7255.
[12] Hazelrigg, G. A., 1998, "A Framework for Decision-Based Engineering Design," J. Mech. Des., **120**(4), pp. 653–658.
[13] Balling, R., 1999, "Design by Shopping: A New Paradigm?," Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization WCSMO-3, Buffalo, NY, University at Buffalo, pp. 295–297.

[14] Winer, E. H., and Bloebaum, C. L., 2001, "Visual Design Steering for Optimization Solution Improvement," Struct. Multidisc. Optim., **22**(3), pp. 219–229.
[15] 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," Struct. Multidisc. Optim., **23**(6), pp. 412–424.
[16] Brill, E. D., Jr., 1979, "The Use of Optimization Models in Public-Sector Planning," Manage. Sci., **25**(5), pp. 413–422.
[17] Chang, S-Y., Brill, E. D., Jr., and Hopkins, L. D., 1983, "Modeling to Generate Alternatives: A Fuzzy Approach," Fuzzy Sets Syst., **9**(1–3), pp. 137–151.
[18] Huang, G. H., Gaetz, B. W., and Patry, G. G., 1996, "A Grey Hop, Skip, and Jump Approach: Generating Alternatives for Expansion Planning of Waste Management Facilities," Can. J. Civil Eng., **23**(6), pp. 1207–1219.
[19] DeCarolis, J. F., 2011, "Using Modeling to Generate Alternatives (MGA) to Expand Our Thinking on Energy Futures," Energy Econ., **33**(2), pp. 145–152.
[20] Nunez, M., Datta, V. C., Molina-Cristobal, A., Guenov, M., and Riaz, A., 2012, "Enabling Exploration in the Conceptual Design and Optimisation of Complex Systems," J. Eng. Des., **23**(10–11), pp. 849–872.
[21] Jeon, Y. H., Jun, S., Kang, S., and Lee, D. H., 2012, "Systematic Design Space Exploration and Rearrangement of the MDO Problem by Using Probabilistic Methodology," J. Mech. Sci. Technol., **26**(9), pp. 2825–2836.
[22] Rahman, S., and Shrestha, G., 1992, "A Technique to Incorporate New Information in Evaluating Generation Alternatives," IEEE Trans. Power Syst., **7**(2), pp. 900–906.
[23] Daniel, H. L., Ranjithan, S. R., Brill, E. D., Jr., and Baugh, J. W., Jr., 2001, "Genetic Algorithm Approaches for Addressing Unmodeled Objectives in Optimization Problems," Eng. Optim., **33**(5), pp. 549–569.
[24] Neches, R., and Madni, A. M., 2012, "Towards Affordably Adaptable and Effective Systems," Syst. Eng., **16**, pp. 224–234.
[25] Simpson, T. W., and Martins, J. R., 2011, "Multidisciplinary Design Optimization for Complex Engineered Systems: Report From a National Science Foundation Workshop," ASME J. Mech. Des., **133**(10), p. 101002.
[26] Madni, A. M., Brenner, M. A., Costea, I., MacGregor, D., and Meshkinpour, F., 1985, "Option Generation: Problems, Principles, and Computer-Based Aiding," Proceedings of the 1985 International Conference on Systems, Man, and Cybernetics, Tuscon, AZ, pp. 757–760.
[27] Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York, NY.
[28] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Trans. Evol. Comput, **6**(2), pp. 182–197.
[29] Deb, K., 2009, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, New York, NY.
[30] Mackenna, A., 2011, "Rapid Ship Design Environment," Physics-Based Modeling in Design and Development for U.S. Defense Conference," Denver, CO, Paper No. 13623.
[31] Simpson, T. W., Rosen, D., Allen, J. K., and Mistree, F., 1996, "Metrics for Assessing Design Freedom and Information Certainty in the Early Stages of Design," Proceedings of the 1996 ASME DETC and CIE, Irving, CA, Paper No. 96-DETC/DTM-1521.
[32] Finch, W. W., and Ward, A. C., 1997, "A Set-Based System for Eliminating Infeasible Designs in Engineering Problems Dominated by Uncertainty," Proceedings of the 1997 ASME Design Engineering Technical Conferences, Sacramento, CA, Paper No. DETC97/DTM-3886.
[33] Chen, W., and Lewis, K., 1999, "A Robust Design Approach for Achieving Flexibility in Multidisciplinary Design," AIAA J., **37**, pp. 982–989.
[34] Shan, S., and Wang, G. G., 2004, "Space Exploration and Global Optimization for Computationally Intensive Design Problems: a Rough Set Based Approach," Struct. Multidisc. Optim., **28**(6), pp. 427–441.
[35] Madhavan, K., Shahan, D., Seepersad, C. C., Hlavinka, D. A., and Benson, W., 2008, "An Industrial Trial of a Set-Based Approach to Collaborative Design," Proceedings of the IDETC/CIE 2008, New York, NY, Paper No. DETC2008/49953.
[36] Campbell, M., "The Skewboid Method: A Simple and Effective Approach to Pareto Relaxation and Filtering," 2012, ASME IDETC, Chicago, IL, Paper No. DETC2012-70323.
[37] Messac, A., Martinez, M. P., and Simpson, T. W., 2002, "Introduction of a Product Family Penalty Function Using Physical Programming," ASME J. Mech. Des., **124**(2), pp. 164–172.
[38] Chen, W., Sahai, A., Messac, A., and Sundararaj, G. J., 2000, "Exploration of the Effectiveness of Physical Programming in Robust Design," ASME J. Mech. Des., **122**(2), pp. 155–163.
[39] Liu, H., Chen, W., Scott, M. J., and Qureshi, K., 2008, "Determination of Ranged Sets of Design Specifications by Incorporating Design-Space Heterogeneity," Eng. Optim., **40**(11), pp. 1011–1029.
[40] Bertsche, B., 2008, *Reliability in Automotive and Mechanical Engineering: Determination of Component and System Reliability*, Springer-Verlag, New York, NY.
[41] Gunawan, S., and Azarm, S., 2005, "A Feasibility Robust Optimization Method Using Sensitivity Region Concept," J. Mech. Des., **127**(5), pp. 858–865.
[42] Li, M., Williams, N., and Azarm, S., 2009, "Interval Uncertainty Reduction and Single-Disciplinary Sensitivity Analysis With Multi-Objective Optimization," J. Mech. Des., **131**(3), p. 031007.
[43] Eddy, J., and Lewis, K., 2002, "Visualization of Multi-Dimensional Design and Optimization Data Using Cloud Visualization," Proceedings of the 2002 ASME Design Engineering Technical Conferences, Montreal, Canada, Paper No. DETC2002/DAC-34130.
[44] Shneiderman B., 1992, "Tree Visualization with Tree-Maps: 2-D Space-Filling Approach," ACM Trans. Graph., **11**(1), pp. 92–99.
[45] "Honeycomb^{TM}," http://www.hivegroup.com/
[46] Chien, S.-F., and Flemming, U., 2002, "Design Space Navigation in Generative Design Systems," Autom. Constr., **11**(1), pp. 1–22.
[47] Aksit, M., and Marcelloni, F., 2001, "Deferring Elimination of Design Alternatives in Object-Oriented Methods," Concurrency Computation: Pract. Exper., **13**(14), pp. 1247–1279.
[48] Ahn, B. S., and Park, H., 2008, "An Efficient Pruning Method for Decision Alternatives of OWA Operators," IEEE Trans. Fuzzy Syst., **16**(6), pp. 1542–1549.
[49] Stump, G. M., Yukish, M., and Merenich, J. J., 2005, "Tracing Interesting Features in Trade Spaces to Conceptual Model Design Rules," 2005 IEEE Aerospace Conference, Vols. 1–4, pp. 4227–4238.
[50] Malakooti, B., and Raman, V., 2000, "Clustering and Selection of Multiple Criteria Alternatives Using Unsupervised and Supervised Neural Networks," J. Intell. Manuf., **11**(5), pp. 435–451.
[51] Chen, Y., Kilgour, D. M., and Hipel, K. W., 2012, "A Decision Rule Aggregation Approach to Multiple Criteria-Multiple Participant Sorting," Group Decis. Negotiation, **21**(5), pp. 727–745.
[52] Malak, R., 2008, "Using Parameterized Efficient Sets to Model Alternatives for Systems Design Decisions," Ph.D. thesis, Georgia Tech, Atlanta, GA.
[53] Kurtoglu, T., and Campbell, M. I., 2009, "An Evaluation Scheme for Assessing the Worth of Automatically Generated Design Alternatives," Res. Eng. Des., **20**(1), pp. 59–76.
[54] Kim, J. K., and Kim, T. G., 2006, "A Plan-Generation-Evaluation Framework for Design Space Exploration of Digital Systems Design," IEICE Trans. Fundam. Electron., Commun. Comput. Sci., **E89-A**(3), pp. 772–781.
[55] Halim, I., and Srinivasan, R., 2008, "Designing Sustainable Alternatives for Batch Operations Using an Intelligent Simulation-Optimization Framework," Chem. Eng. Res. Des., **86**(7A), pp. 809–822.
[56] Smith, G., Richardson, J., Summers, J. D., and Mocko, G. M., 2012, "Concept Exploration Through Morphological Charts: An Experimental Study," ASME J. Mech. Des., **134**(5), p. 051004.
[57] Krishnan, V., and Katkoori, S., 2006, "A Genetic Algorithm for the Design Space Exploration of Datapaths During High-Level Synthesis," IEEE Trans. Evol. Comput., **10**(3), pp. 213–229.
[58] Collignan, A., Sebastian, P., Pailhes, J., and Ledoux, Y., 2012, "Arc-Elasticity and Hierarchical Exploration of the Neighborhood of Solutions in Mechanical Design," Adv. Eng. Inf., **26**(3), pp. 603–617.
[59] Shir, O., Preuss, M., Naujoks, B., and Emmerich, M., 2009, "Enhancing Decision Space Diversity in Evolutionary Multiobjective Algorithms," Evol. Multi-Criterion Optim., **5467**, pp. 95–109.
[60] Toffolo, A., and Benini, E., 2003, "Genetic Diversity as an Objective in Multi-Objective Evolutionary Algorithms," Evol. Comput., **11**(2), pp. 151–167.
[61] Tamaskar, S., Neema, K., Kotegawa, T., and DeLaurentis, D., 2011, "Complexity Enables Design Space Exploration," Conference Proceedings—IEEE International Conference on Systems, Man, and Cybernetics, pp. 1250–1255.
[62] Kang, E., Jackson, E., and Schulte, W., 2011, "An Approach for Effective Design Space Exploration," Lect. Notes Comput. Sci., **6662**, pp. 33–54.
[63] Ferguson, S., Gurnani, A., Donndelinger, J., and Lewis, K., 2005, "A Study of Convergence and Mapping in Multibojective Optimization Problems," Int. J. Veh. Syst. Model. Test., **1**(1/2/3), pp. 192–215.
[64] Simon, P. R., and Ferguson, S., 2010, "Investigating the Significance of 'One-to-Many' Mappings in Multiobjective Optimization," 2010 ASME Design Engineering Technical Conferences, Montreal, Quebec, Paper No. DETC2010-28689.
[65] Brill, E. D., Chang, S.-Y., and Hopkins, L. D., 1982, "Modeling to Generate Alternatives: The HSJ Approach and an Illustration Using a Problem in Land Use Planning," Manage. Sci., **28**(3), pp. 221–235.
[66] Kripakaran, P., and Gupta, A., 2006, "MGA–A Mathematical Approach to Generate Design Alternatives," Lect. Notes Comput. Sci., **4200**, pp. 408–415.
[67] Zechman, E. M., and Ranjithan, R. S., 2007, "Generating Alternatives Using Evolutionary Algorithms for Water Resources and Environmental Management Problems," J. Water Resour. Plann. Manage., **133**(2), pp. 156–165.
[68] Foster, G., 2011, "Expansion of Alternative Generation Techniques," Master's thesis, North Carolina State University, Raleigh, NC.
[69] Foster, G., and Ferguson, S., 2010, "Exploring the Impact of Distance Metrics on Alternative Generation in a Multiobjective Problem," 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX, Paper No. AIAA-2010-9091.
[70] Foster, G., and Ferguson, S., 2011, "Assessing the Effectiveness of Using Graveyard Data for Generating Design Alternatives," Proceedings of the 2011 ASME Design Engineering Technical Conference, Washington, DC, Paper No. DETC2011-48636.
[71] Sofge, D. A., 2002, "Using Genetic Algorithm Based Variable Selection to Improve Neural Network Models for Real-World Systems," Proceedings of the 2002 International Conference on Machine Learning and Applications.
[72] Simpson, T. W., Poplinski, J. D., Koch, P. N., and Allen, J. K., 2001, "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," Eng. Comput., **17**(2), pp. 129–150.
[73] Turner, C. J., 2011, "Metamodeling in Product and Process Design," Proceedings of the 2011 IDETC and CIE, Washington, DC, Paper No. DETC2011-47483.

[74] Viana, F. A. C., Gogu, C., and Haftka, R. T., 2010, "Making the Most Out of Surrogate Models: Tricks of the Trade," Proceedings of the 2010 ASME IDETC and CIE, Montreal, Canada, Paper No. DETC2010-28813.
[75] The MathWorks, Inc., "Find Minimum of Function Using Pattern Search—MATLAB," http://www.mathworks.com/help/toolbox/gads/patternsearch.html
[76] Audet, C., and Dennis, J. E., Jr., 2006, Analysis of Generalized Pattern Searches, DTIC Document.
[77] Kolda, T. G., Lewis, R. M., and Torczon, V., 2003, "Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods," SIAM Rev., **45**, pp. 385–482.
[78] The MathWorks, Inc., "Genetic Algorithm Options: Options Reference (Global Optimization Toolbox)," http://www.mathworks.com/help/toolbox/gads/f6174dfi10.html
[79] Azarm, S., Reynolds, B. J., and Narayanan, S., 1999, "Comparison of Two Multiobjective Optimization Techniques With and Within Genetic Algorithms," Proceedings of the 1999 ASME Design Engineering Technical Conferences, Las Vegas, NV, Paper No. DETC99/DAC-8584.
[80] Hacker, K., and Lewis, K., 2002, "Robust Design Through the Use of a Hybrid Genetic Algorithm," Proceedings of the 2002 DETC and CIE Conference, Montreal, Canada, Paper No. DETC2002/DAC-34108.
[81] The MathWorks, Inc., "Find Minimum of Unconstrained Multivariable Function Using Derivative-Free Method—MATLAB," http://www.mathworks.com/help/techdoc/ref/fminsearch.html
[82] The MathWorks, Inc., "Find Minimum of Function Using Genetic Algorithm—MATLAB," http://www.mathworks.com/help/toolbox/gads/ga.html
[83] The MathWorks, Inc., "Find Minimum of Constrained Nonlinear Multivariable Function—MATLAB," http://www.mathworks.com/help/toolbox/optim/ug/fmincon.html
[84] Tang, B., 1993, "Orthogonal Array-Based Latin Hypercubes," J. Am. Stat. Assoc., **88**(424), pp. 1392–1397.
[85] Nievergelt, J., 2000, "Exhaustive Search, Combinatorial Optimization and Enumeration: Exploring the Potential of Raw Computing Power," Lect. Notes Comput. Sci., **1963**, pp. 18–35.