

# A constraint-based approach to feasibility assessment in preliminary design

ASHWIN GURNANI,<sup>1</sup> SCOTT FERGUSON,<sup>1</sup> KEMPER LEWIS,<sup>1</sup> AND JOSEPH DONNDELINGER<sup>2</sup>

<sup>1</sup>Design of Open Engineering Systems Laboratory, University at Buffalo–SUNY, Buffalo, New York, USA

<sup>2</sup>Vehicle Development Research Lab, General Motors Research & Development Center, Warren, Michigan, USA

(RECEIVED October 3, 2005; ACCEPTED July 6, 2006)

## Abstract

In this paper, we present the development and application of a technical feasibility model used in preliminary design to determine whether a set of desired product specifications obtained from marketing is feasible in the engineering domain. This model is developed by integrating the capabilities of a multiobjective design problem, a multicriteria design optimization tool, a Pareto frontier gap analyzer, metamodeling methods, and use of the Pareto frontier as a constraint for feasibility assessment. Although the tools are independent of the domain, their application is illustrated using two examples: a simple three-objective mathematical problem and a five-objective passenger vehicle design problem. The feasibility of the desired product specifications is determined with respect to the problem's Pareto frontier, which is considered to be the necessary *constraint* to satisfy.

**Keywords:** Decision Making; Decision Support System; Multiobjective Optimization; Pareto Frontier

## 1. INTRODUCTION

The goal of this research is to develop analytical tools to support the definition of a feasible and technically superior set of product specifications in the preliminary stages of the product development process. Ensuring that product specifications are mutually compatible and feasible from an engineering perspective is not a trivial task. It is, however, a paramount task in effective execution of the product development process. The approach developed and presented in this paper occurs at the intersection of two technology areas: the development and application of multidisciplinary design, and optimization decision-support systems and the generation of Pareto frontiers to be used as constraints for feasibility assessment within multiobjective optimization problems.

Decision-support systems for the design of large, complex engineering systems are critical to making accurate and appropriate decisions in a timely fashion. These decision-support systems are often embodied in software tools that provide for substantial automation of the design process

and for the application of multidisciplinary and multiobjective optimization methods. The popularity of these frameworks has given rise to a number of research programs within various sectors of the engineering design community.

In the academic sector, the Center for Research in Computation and its Applications has developed the Virtual Airplane Design and Optimization framework (Alzubbi et al., 2000). In the government sector, Sandia National Labs has developed the Design and Analysis Kit for Optimization (Eldred et al., 2004), and researchers at NASA Langley have developed the Framework for Interdisciplinary Optimization (Townsend et al., 1998). In the private sector, frameworks have been developed both as commercial products and as proprietary applications. Commercial products include the GENESIS/SDRC I-DEAS solution (Kosaka et al., 2000) offered by Vanderplaats R&D and the commercial version of the Federated Intelligent Product Environment (Röhl et al., 2000) being developed by Engineous software. Proprietary corporate solutions include Ford Motor Company's design synthesis system based on the quantum leap adaptive optimization engine (Port, 1998) and the multidisciplinary design frameworks developed within the General Motors (GM) R&D Center (Bennett et al., 1995; Fenyés et al., 2002). In recent work, customer preferences have been gathered using a Web-based framework and integrated into the design and

Reprint requests to: Kemper Lewis, Design of Open Engineering Systems Laboratory, University at Buffalo–SUNY, Buffalo, NY 14260, USA.  
E-mail: kelewis@eng.buffalo.edu

manufacturing process of a customized product (Siddique & Boddu, 2004). All the different multidisciplinary frameworks discussed here provide tools to incorporate system constraints and determine solutions that are feasible with respect to these constraints.

These frameworks provide powerful engineering analysis environments with substantial improvements in operational efficiency; however, they are often computationally very expensive, especially when they include finite element analyses. One means of further improving computational speed is to employ surrogate models in place of the computationally expensive models. Surrogate models have been applied successfully within the automotive industry in robust engineering applications (Lee et al., 2001) and in the Multi-function Optimization Visual Environment (Longacre et al., 1996). The approach in the latter is to employ a series of discipline-specific surrogate models with a coordinated schema of design variables and responses. In contrast, the approach outlined in this paper is to exercise a multidisciplinary design framework to generate a single surrogate model. This surrogate model is a representation of the Pareto frontier for a problem with multiple desired performance attributes. This representation of the Pareto frontier also forms the constraint against which feasibility of new designs is evaluated.

Developing design frameworks using constraints is not new to engineering design. Constraints have been developed from decision-maker preferences and used to determine attribute weights in multiattribute selection problems (See et al., 2004). In addition, robust design alternatives have been obtained for multiattribute selection problems through the use of intelligent sampling of the design space and incorporating preference based constraints in the selection methodology (Gurnani & Lewis, 2005). Constraint satisfaction problems have been formulated and used in design, integrating a function-based design generation algorithm to a geometric modeler (Schmidt et al., 2005). A formalism for operational representation of constraints for feature-based design has been proposed by Fu and de Pennington (1993), and Marinescu and Dechter (2005) have presented advanced approaches for investigation of the design space in discrete variable constraint optimization problems.

In addition to the use of constraints in developing design frameworks, this work uses constraints within surface fitting techniques to determine a representation of the Pareto surface. Goel et al. (2004) have used a response surface method along with an evolutionary algorithm and constraint strategy to approximate the Pareto optimal front and apply it to rocket injector design. Kasprzak and Lewis (2000) have also used curve-fitting techniques to determine a Pareto front and use it to aid designers in making trade-offs in decisions pertaining to solutions along the Pareto frontier. Although constraints have been used in design and optimization in a number of effective ways including developing approximations of Pareto frontiers, the use of an approximated Pareto frontier as a constraint in feasibility assess-

ment is the novel and primary development and contribution of this paper. In addition, the developed methodology is for the preliminary design stage within the design process. Although multiobjective optimization techniques are used to generate the Pareto surface, this paper does not prescribe any type of optimization per se. The premise of this paper is determining the engineering feasibility of new combinations of objective functions known to maximize customer satisfaction as obtained by marketing. Engineering feasibility refers to the ability of engineering to actually design a product with the desired performance characteristics. The feasibility is checked for *yet to be developed* products (hence, preliminary design) with respect to a preexisting analysis system.

To create the surrogate model for the Pareto frontier in this paper, we use a multiobjective genetic algorithm (MOGA) in conjunction with a multiobjective analyses system to generate a set of Pareto optimal points. A gap analyzer has also been developed to identify and populate any gaps that might exist within the Pareto set by tuning the MOGA to populate these gaps. An effective representation of the Pareto frontier is then determined using metamodeling techniques. Once a complete representation of the Pareto frontier is developed, it can be viewed as the *constraint* in order to determine if a proposed vehicle is

- *feasible*, if the vehicle is inside the Pareto frontier;
- *feasible and technically superior*, if the vehicle is on the Pareto frontier; or
- *infeasible*, if the vehicle is outside the Pareto frontier.

This information may then be used by engineers to identify compatible combinations of performance targets and design parameters. Some fundamentals of multiobjective optimization are presented in the next section. Section 3 presents a detailed discussion on the development of the different stages needed in the construction of a constraint-based technical feasibility model (TFM), building upon the concepts presented in Section 2. Section 4 discusses the development of a TFM for a three-objective mathematical problem and a passenger car design problem and also presents the results of the process. The final section of this paper covers the conclusions of the results as well as recommendations for future work in this area.

## 2. PARETO FRONTIER GENERATION

Multiobjective optimization includes a set of formal tools aimed at providing designers with accurate, complete, and rational information to make effective decisions in engineering. Fundamental to multiobjective optimization in general and to this paper in particular is the concept of Pareto optimality. When multiple competing objectives or *criteria* exist, the optimum is no longer a single design point but an entire set of nondominated design points. This is commonly known as the Pareto set (Pareto, 1896, 1906/1971). The

Pareto set is composed of Pareto optimal solutions. In simple terms, a Pareto optimal solution is one for which any improvement in one objective must result in the degradation of at least one other objective. Mathematically, a feasible design variable vector ( $\bar{x}'$ ) is Pareto optimal if and only if there is no feasible design variable vector ( $\bar{x}$ ), with the characteristics,

$$\begin{aligned} f_i(\bar{x}) &\leq f_i(\bar{x}') \quad \forall i, \quad i = 1, \dots, n \\ f_i(\bar{x}) &< f_i(\bar{x}') \quad \text{for at least one } i, \quad i = 1, \dots, n \end{aligned} \quad (1)$$

where  $n$  is the number of objectives and the use of the less than symbol indicates an improvement in the objective function (assuming minimization of objectives is desired). The Pareto set can be used to generate a Pareto frontier, a continuous mathematical function representing all of the possible Pareto optimal solutions.

Traditional methods of generating Pareto frontiers employ repeated conversion of multiobjective problems into single objective problems. However, these methods are contradictory to the spirit of Pareto optimality, and have been proven to perform poorly when attempting to populate Pareto frontiers under many circumstances (Koski, 1985; Das & Dennis, 1997; Messac et al., 2000). To avoid these problems, many researchers have turned to other methods for generating Pareto frontiers. Messac and Sundararaj (2000) have applied physical programming to generate Pareto frontiers without relying on weights, instead using designer preferences in the form of metric classes in the optimization process. Narayanan and Azarm (1999) have developed the interactive sequential hybrid optimization technique, solving a multiobjective problem through the repeated application of a simple genetic algorithm. A discussion of the trade-offs in multiobjective optimization when using operators with and within genetic algorithms can be found in (Azarm et al., 1999). Fitness and ranking schemes for uses in genetic programming are developed by Balling (2000), Goldberg (1989), and Fonseca and Fleming (1998), whereas the development of genetic programming to efficiently generate a thorough spread of points along a Pareto frontier is found in the paper by Eddy and Lewis (2001). Genetic programming has also been used as a tool for invention and creativity in automated design (Koza et al., 2004) and applied to the design of combinational logic circuits (Coello & Aguirre, 2002).

Given this background, the approach taken in this paper for solving multiobjective continuous problems involves

the use of genetic programming. A MOGA can be used to populate the entire Pareto frontier in a single optimization run without repeated conversion from a multiobjective to a single objective problem. In a multiobjective problem, there is not a single measure of performance and a simple greater than/less than comparison is no longer sufficient. The fitness evaluation in a MOGA incorporates the concept of Pareto optimality such that a design that exhibits dominant performance characteristics is favored over one that does not and is therefore more likely to proliferate. In addition, MOGAs typically require far fewer function evaluations to converge to a set of solutions than other methods (e.g., grid searches, iterative weighted sums; Goldberg, 1989; Davis, 1991; Miller & Goldberg, 1996). Another advantage is that the MOGA is very robust to ill-conditioned problems (multimodal, discontinuous, discrete, etc.; Deb & Kumar, 1995). Because of this, the MOGA is more likely than other methods to yield a dense and uniformly populated Pareto frontier (Fonseca & Fleming, 1995; Kurapati et al., 2002). Furthermore, because the MOGA is of zero order, the form of the evaluation function is irrelevant to the workings of the algorithm and the method therefore lends itself well to use with other analysis codes for which there is no analytical evaluation function. Also of importance, a MOGA can be tailored for specific problems with computational complexity and parallel computing issues in mind (Stender, 1993; Lee & Hajela, 1996). The MOGA used in this paper is one adapted from Eddy and Lewis (2001), and is currently also being used in the Design and Analysis Kit for Optimization tool for optimization and analysis developed at Sandia National Laboratories (Eldred et al., 2004).

Having introduced the key technology areas that provide the background of this paper, it is now possible to begin tying them together into an effective framework. The framework used is constraint based in that it uses the Pareto frontier as a constraint in order to determine feasibility of new test designs. The next section discusses the development of such a framework and the background behind each block.

### 3. TECHNICAL FEASIBILITY MODEL FRAMEWORK

The framework of the TFM is shown in Figure 1. The first step is the identification and formulation of the engineering system that is being investigated. For this system, an effective and efficient MOGA is then developed and tuned to

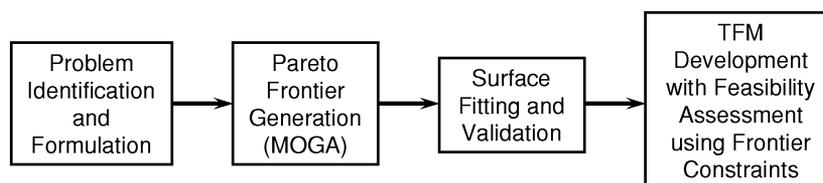


Fig. 1. The technical feasibility model development process.

populate the Pareto frontier. Once the Pareto set has been sufficiently populated, a mathematical representation of it is developed using metamodeling techniques to create the Pareto frontier. Validation techniques are then applied to the fit surface to ensure that the correct behavior of the system is being modeled. This forms the third step of the TFM. The final step in the TFM development is the integration of different feasibility assessment tools using the fitted Pareto surface as a constraint, in turn providing the user with feasibility information of given test points.

The following sections detail the different blocks of the TFM and illustrate the solution for each step using a simple three-objective problem. Section 4 provides results of a benchmarking feasibility analysis on a cross-section of passenger vehicles currently on the market. The results of this feasibility assessment are based upon the problem formulated by GM R&D.

### 3.1. Problem formulation

For the purpose of illustration, the approach to determine technical feasibility is applied to a multiobjective problem adapted from Viennet et al. (1996). This problem consists of three-objective functions and two design variables, suiting both the needs for an illustrative example and in the ability to visualize the results. The problem statement is given in Eq. (2).

$$\begin{aligned} \text{minimize: } & F_1 = x^2 + (y - 1)^2, \\ & F_2 = x^2 + (y + 1)^2 + 1, \\ & F_3 = (x - 1)^2 + y^2 + 2; \\ \text{subject to: } & -2 \leq x, \quad y \leq 2. \end{aligned} \quad (2)$$

Once the problem is identified, the Pareto set is obtained using a MOGA. This is discussed in the next section.

### 3.2. Frontier generation and gap analyzer

Although a MOGA is very efficient at populating a Pareto set, it may not always cover the entire performance space. This is most likely to occur when exercising systems in which an evaluation of a design point is either computationally or time expensive. In these cases, parts of the Pareto frontier have been identified as seen in Figure 2, but there exist significant gaps in the frontier that may decrease the fidelity of the surface fit to the frontier. To obtain a complete representation of the Pareto frontier, further system analyses are required, which significantly increases the computational expense. The utopia point, which represents the individual best values for each objective, is also marked in Figure 2. (Note: the identified parts of the Pareto frontier in Fig. 2 have hashed marks on one side to indicate that the Pareto frontier is used as a constraint for feasibility assessment. Feasibility assessment is discussed later in the paper.)

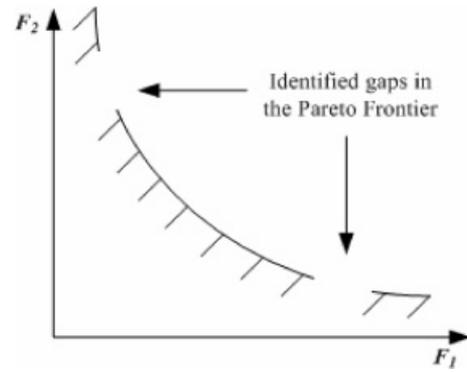


Fig. 2. The initial Pareto frontier from multiobjective genetic algorithm results.

To overcome running a large number of system analyses to achieve a *gap free* Pareto surface, a gap analyzer has been developed to work with the MOGA in populating these gaps and ensuring the integrity of the surface fit. The central concept of the gap analyzer is to identify gaps in the Pareto set and then run localized MOGAs in these gaps to populate them. For the gap analyzer, the performance space is discretized into a series of discrete boxes, with step sizes specified by the designer. This allows the designer to select meaningful sizes for the gaps in the Pareto frontier that will be identified. For example, consider the objective of 0–60 mph acceleration time. A step size of 1/20 of a second would be too fine; it would result in identification of a large number of gaps, along with a dramatic increase in either computational or run-time expense, but this increment is practically imperceptible by most customers. Likewise, a step size of 2 s would be too coarse; it would substantially reduce the number of gaps identified and the computational expense associated with filling them, but this increment is far beyond the threshold of customer perception. The selection of step size for an objective therefore involves a trade-off between gap identification and the time and resources needed to evaluate populations of points. For the GM problem, designer preferences obtained from GM are used to determine the appropriate step size in this paper.

The discretization of the performance space results in a set of *hyperboxes*. (The discretized regions are called hyperboxes because the performance space can have more than three-objective functions, representing each edge of the discretized region). This set of hyperboxes contains populated (populated with Pareto optimal points) and nonpopulated hyperboxes (containing no points). All hyperboxes (both populated and nonpopulated) are stored in an array, and each hyperbox is labeled by the index number of its corresponding discretized unit along each objective. For example, in a two-objective problem, if a hyperbox is the third discretized unit along one objective and the second discretized unit along the second objective, it is stored as (3, 2) in the array of all hyperboxes. The gap analyzer methodology

uses these indices to determine *weak* and *strong* domination between these hyperboxes in determining gaps in the Pareto set. Before explaining the details of the gap analyzer, definitions of weak and strong domination are presented.

Consider two hyperboxes B and C, where  $f_{B_{\max,i}}$  is the maximum objective function index of hyperbox B for the  $i$ th objective,  $f_{C_{\max,i}}$  is the maximum objective function index of hyperbox C for the  $i$ th objective, and  $n$  is the number of objective functions.

Assuming minimization of objective functions, hyperbox B is said to weakly dominate hyperbox C if all conditions in Eqs. (3)–(5) are satisfied.

$$f_{B_{\max,i}} \leq f_{C_{\max,i}}, \quad \forall i = 1, \dots, n; \quad (3)$$

$$f_{B_{\max,i}} = f_{C_{\max,i}}, \quad \text{for at least one } i; \quad (4)$$

$$f_{B_{\max,i}} < f_{C_{\max,i}}, \quad \text{for at least one } i. \quad (5)$$

Note that if maximization of objective functions is required, the conditions are simply inverted.

For *strong domination*, hyperbox B is said to strongly dominate hyperbox C if Eq. (6) holds true.

$$f_{B_{\max,i}} < f_{C_{\max,i}}, \quad \forall i = 1, \dots, n. \quad (6)$$

Given these definitions for weak and strong domination, the gap analyzer methodology is now presented. The gap analyzer begins with an array representing all hyperboxes that form the performance space and the knowledge of hyperboxes representing the Pareto set (populated hyperboxes). There are four steps for the gap analyzer.

STEP 1. Eliminate all hyperboxes that do not weakly or strongly dominate a populated hyperbox. As seen in Figure 3, the regions shaded in gray correspond to the hyperboxes that do not dominate a populated or Pareto hyperbox. By eliminating these hyperboxes, a large portion of the performance space is eliminated from further consideration effectively reducing the gap search space.

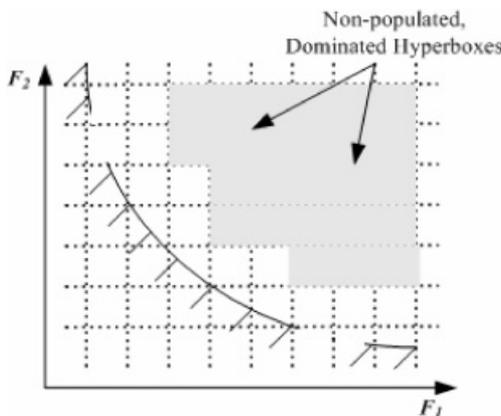


Fig. 3. The elimination of nonpopulated, dominated hyperboxes.

STEP 2. Eliminate all hyperboxes from the discretized performance space that strongly dominate the Pareto hyperboxes. These represent the hyperboxes that form the infeasible region of the performance space and lie below the Pareto set. Figure 4 shows the dominating hyperboxes for a two-dimensional (2-D) problem.

STEP 3. The remaining hyperboxes in the performance space that are not populated correspond to gaps in the frontier. Figure 5 shows the identification of gaps in the frontier from the remaining hyperboxes. To identify the size of each gap, adjacent *gap* hyperboxes are combined into clusters. Clusters with the largest number of hyperboxes correspond to the largest gaps in the frontier and are selected first for investigation.

STEP 4. Run localized MOGA within each identified and clustered gap. The initial population of the MOGA comprises points from the populated hyperboxes adjacent to the gaps. Constraints corresponding to the gap boundaries are placed on the selectors within the MOGA, which drive the process into the gaps and populate points. These new points serve to fill in the gaps in the frontier, ensuring integrity in surface fitting and feasibility assessment.

Note that the figures used in this section to illustrate the gap analyzer algorithm are for a 2-D problem. The implementation of the gap analyzer, however, is not restricted to 2-D problems. The gap analyzer algorithm uses concepts from multiobjective optimization that may be applied for any number of objectives. It is inherently a numerical, iteration-based method that navigates through the performance space, incrementing along one objective at a time.

The gap analyzer is illustrated with the aid of the three-objective example problem next. The initial Pareto frontier is obtained by running a MOGA for 58 generations, resulting in a total evaluation of 10,000 unique designs. These evaluations result in a Pareto frontier composed of 6829 Pareto optimal designs. A large number of designs are chosen to allow for an exhaustive, but not entirely complete

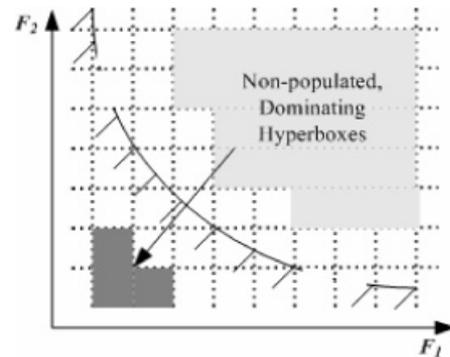


Fig. 4. The identification of nonpopulated, dominating hyperboxes.

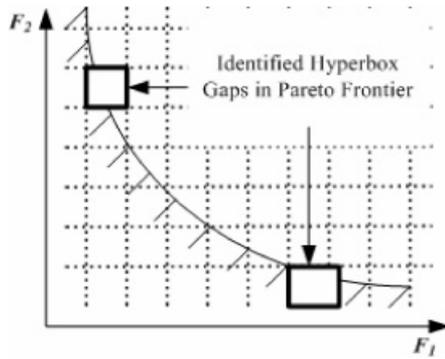


Fig. 5. The identification of gaps in the Pareto surface.

representation of the frontier. The overall shape of the Pareto frontier is studied by plotting the Pareto frontier in the performance space, as shown in Figure 6. The Pareto points appear to be dispersed over the range of the frontier, as there exist no clusters of Pareto points. As stated earlier, the size of the discretization selected for each objective will have a dramatic effect on the number of potential gaps identified in the frontier. For this problem, a discretization size of 0.2 is selected for all objectives. This value is chosen to allow for the identification of gaps that would be of a significant size to warrant further investigation.

Table 1 lists the minimum and maximum values on the Pareto frontier for each objective. Using this information along with the discretization size, the performance space is partitioned into 4831 unique hyperboxes. These correspond to the initial set of hyperboxes representing the performance space required initially for the gap analyzer. Each

Table 1. Maximum and minimum values on the Pareto frontier for each objective

| Objective | Min. Value | Max. Value | Discret. Size |
|-----------|------------|------------|---------------|
| F1        | 0.0041     | 4.0243     | 0.2           |
| F2        | 1.0007     | 5.1950     | 0.2           |
| F3        | 2.0001     | 4.0794     | 0.2           |

hyperbox contains an index for its location in each objective, with index 1 corresponding to the hyperbox index closest to the minimum of that objective. Next, the populated hyperboxes must be identified. As stated in this section, a filled hyperbox is one that has at least one Pareto frontier design located within its bounds. Figure 7 is a plot of the Pareto points in the discretized performance space. The axes relate to the indices of the hyperboxes, and the points correspond to the centroid of each *filled* hyperbox. In this frontier, there are currently 250 points composing the frontier. However, there still are 4581 hyperboxes remaining in the performance space for the gap analyzer analysis, the results of which are discussed next. The results for the three-objective problem are provided for each step of the gap analyzer.

STEP 1. Execution of the gap analyzer on the discretized performance space begins with the identification of hyperboxes that are not populated *and* dominated by Pareto points. Of the 4581 hyperboxes to be accounted for in the performance space, it is found that 3237 of the hyperboxes are

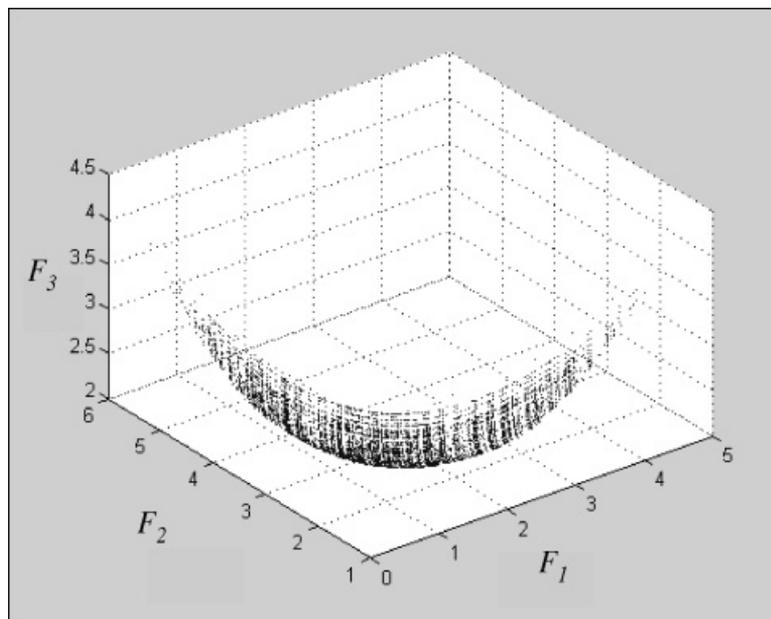


Fig. 6. The Pareto frontier for the three-objective problem from multiobjective genetic algorithm results.

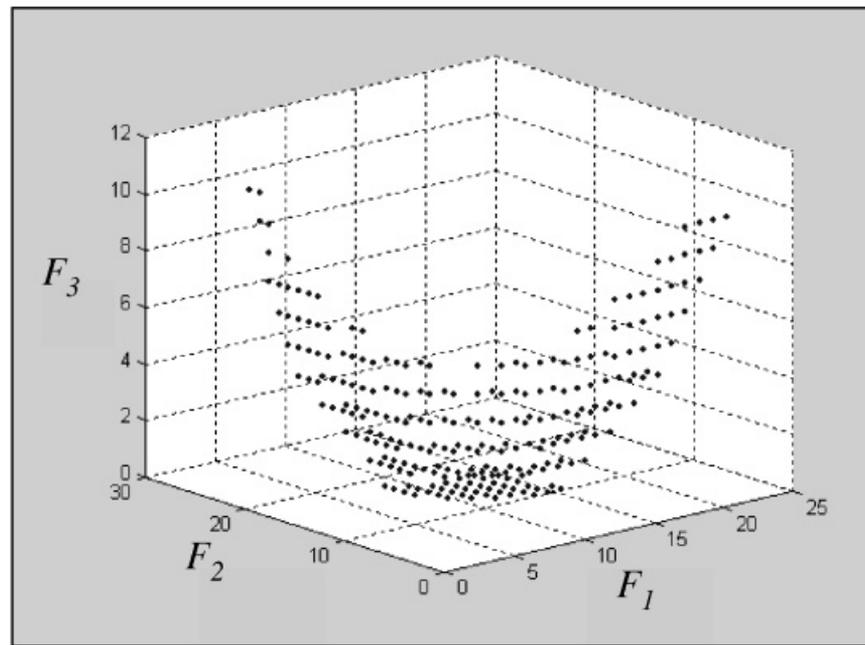


Fig. 7. The distribution of hyperboxes per gap cluster.

strongly dominated by Pareto points. In addition, 764 hyperboxes are found to be weakly dominated by the hyperboxes containing Pareto points.

STEP 2. Next, the hyperboxes that strongly dominate populated hyperboxes are removed, as they comprise the infeasible region of the performance space. There are 370 hyperboxes identified in the performance space that strongly dominate hyperboxes containing Pareto frontier points.

STEP 3. This analysis leaves 230 unique hyperboxes in the performance space that correspond to gaps in the Pareto frontier. The breakdown of the original 4831 hyperboxes into their separate categories is shown in Table 2. As expected, the majority of hyperboxes in the performance space are strongly dominated by filled hyperboxes. The number of hyperboxes needing investigation as potential gaps

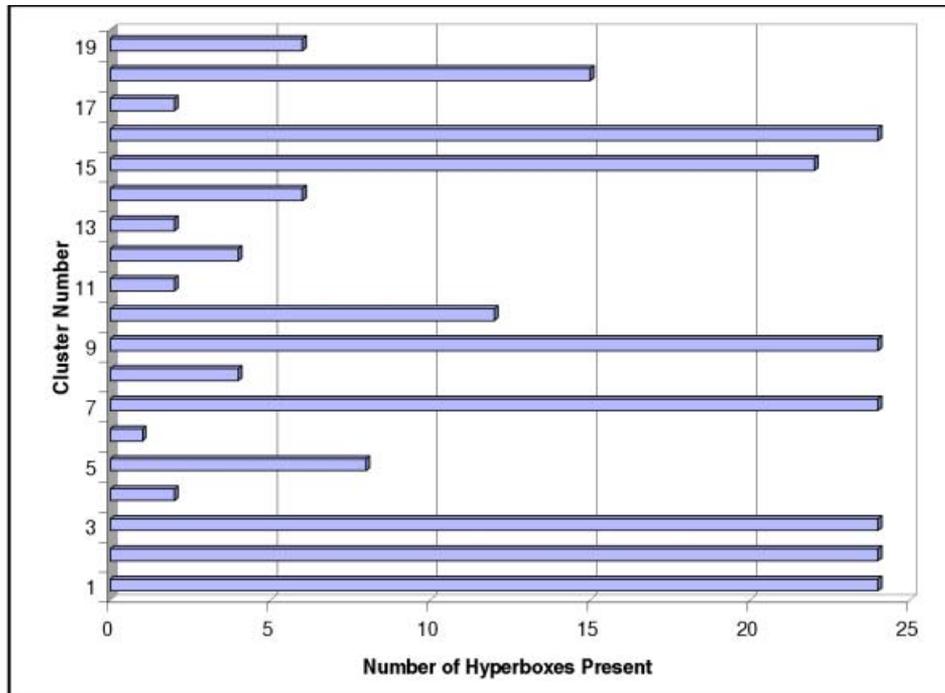
(230 in this problem) relate to only 4.6% of the entire performance space, greatly narrowing the region of space that must be investigated.

Clustering of the gaps is completed by combining hyperboxes that are adjacent to each other in the performance space. For this work, adjacency is defined as having an index number a maximum of one increment away in any objective. Also implemented in this clustering procedure is the constraint that a cluster is stopped after it begins to contain more than 10% of the total number of hyperboxes identified as potential gaps (10% of 230 hyperboxes in this example). This is done in order to prevent combining all hyperboxes identified as gaps into a single cluster. Using this constraint, 19 different clusters were identified in the performance space. The number of hyperboxes found in each cluster is shown in Figure 8.

Table 2. Analysis of hyperbox type and percentage composition in the performance space

| Hyperbox Type   | No. of Hyperboxes | Perform. Space (%) |
|---|-------------------|--------------------|
| Filled (Pareto)   | 250               | 5.0                |
| Strongly dominated by filled hyperboxes                       | 3237              | 67.0               |
| Strongly dominating filled hyperboxes                         | 370               | 7.6                |
| Weakly dominated hyperboxes not adjacent to filled hyperboxes | 764               | 15.8               |
| Remaining hyperboxes adjacent to filled hyperboxes (gaps)     | 230               | 4.6                |
| Total   | 4831              | 100.0              |

STEP 4. Next, instances of the MOGA are created to attempt to place designs within the 19 clusters. As this problem is a minimization problem, the upper index along every objective function of each cluster is used to constrain the upper bound of each objective function. This ensures that the MOGA will attempt to populate only the region of interest as defined by each cluster. As an example, two clusters are used in the following discussion. Clusters 1 and 4 have been selected, and the upper bound of the indices for each cluster is shown in Table 3. Each instance of the MOGA is initialized and allowed to run for 1000 evaluations for the gap clusters. This number is arbitrarily selected because the goal of this exercise is not computational efficiency, but instead determining if a selected gap can be filled. Gap 1 is



**Fig. 8.** A hyperbox index representation of the Pareto frontier. [A color version of this figure can be viewed online at [www.journals.cambridge.org](http://www.journals.cambridge.org)]

known to consist of 24 clustered hyperboxes in the performance space, and after evaluating the feasible designs found from the local MOGA run, two of those hyperboxes are filled with designs. Gap 4 is composed of only two hyperboxes, and the separate instance of the MOGA is unable to find any designs that would fill those hyperboxes. This data is summarized in Table 4. The filling of gaps 1 and 4 is for illustration purposes, and the additional points do not significantly impact the developed Pareto frontier. This is because for this problem, the function evaluations are cheap and an exhaustive MOGA provides a sufficiently populated Pareto set.

Having filled hyperboxes while studying gap 1, those hyperboxes need to be made a part of the Pareto frontier. Modifying Figure 7, the two black circles in the performance space shown in Figure 9 represent the location of the

two hyperboxes filled when studying gap 1. Thus, studying each of the gaps in this manner will provide a complete representation of the Pareto frontier.

An important issue that arises in the Pareto set generation is determining how many points will be sufficient for surface fitting. This is because, for the three-objective example problem presented in this section, function evaluations are cheap (in terms of computational time) and it is possible to run an exhaustive MOGA to determine the entire Pareto set. However, as function evaluations get complicated, the computational time required for an exhaustive MOGA rapidly increases. Therefore, it is necessary to determine the minimum number of Pareto optimal points that would be needed to effectively perform surface fitting and feasibility assessment, which is the focus of Ferguson et al. (2005).

Given a sufficient set of Pareto points, the next step of the TFM is fitting a surface and developing a mathematical representation of the Pareto frontier. This is discussed next in Section 3.3.

**Table 3.** Upper bound indices for selected clusters

| Cluster | Index for |    |    |
|---------|-----------|----|----|
|         | F1        | F2 | F3 |
| 1       | 2         | 19 | 11 |
| 4       | 3         | 11 | 5  |

**Table 4.** Results of gap study

| Cluster | Hyperboxes in Cluster | Filled Hyperboxes |
|---------|-----------------------|-------------------|
| 1       | 24                    | 2                 |
| 4       | 2                     | 0                 |

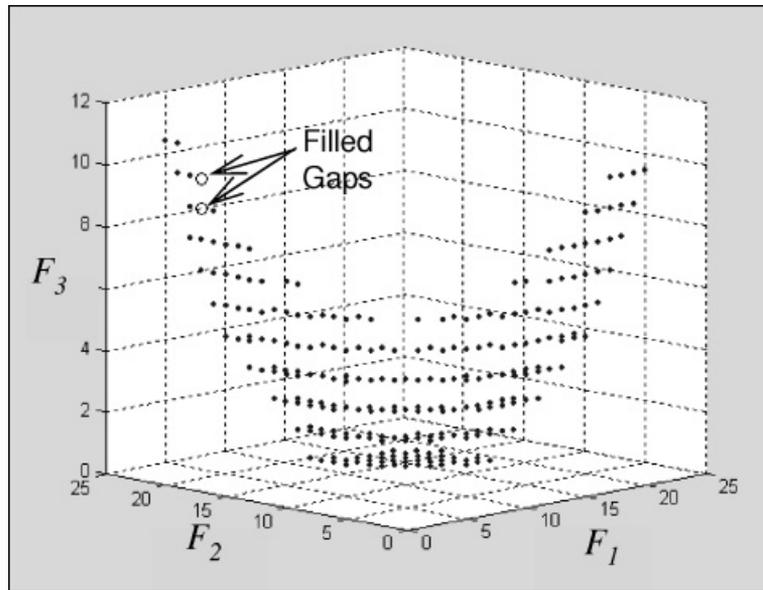


Fig. 9. Filled hyperboxes from the gap 1 study.

### 3.3. Surface fitting and validation

Once the sufficient set of Pareto optimal points has been created, the next step is to develop a means of using it as a constraint in determining whether a given test point is feasible, feasible and Pareto optimal, or infeasible. A test point refers to a particular combination of performance measures that are desired from the product and specified by the marketing domain. The TFM is developed for preliminary design because it determines whether a design with performance measures requested by marketing is *even* feasible in the engineering domain. Because we are using the Pareto set as a constraint for feasibility assessment, the Pareto frontier is henceforth referred to as the *Pareto frontier constraint*. Feasibility determination is accomplished by formulating the Pareto frontier constraint as a continuous mathematical function representing the discrete points in the Pareto set. It is known that for any given point in the performance space, a test point is feasible if it lies above this frontier constraint (assuming minimization of all objectives); it is feasible and Pareto optimal if it lies on the frontier constraint; and it is infeasible if it lies below the frontier constraint.

The region below the frontier constraint is said to be infeasible because the multiobjective nature of the problem does not let the region to be populated. It is important to note that for any multiobjective problem, the Pareto set is the best possible solution set, as mentioned previously, and no design can exist beyond the region constrained by this set. Given the mathematical representation of the Pareto frontier constraint and a single test point, it is possible to determine if the point is feasible with respect to the objectives and system constraints within the engineering domain.

In addition, it is noted that because the analyses are performed by a black box, the system constraints are not directly available for evaluation. Figure 10 depicts the problem statement in two dimensions. As mentioned previously, one side of the Pareto frontier in Figure 10 is hashed to represent the infeasible region of the performance space.

To generate a mathematical representation of the frontier constraint, different model forms are used as bases for fitting the Pareto set to continuous functions against which feasibility of new designs can be tested. The two models investigated in this work are the following:

- a. Unconstrained quadratic function: a second-order equation with no interaction terms is used as the basis func-

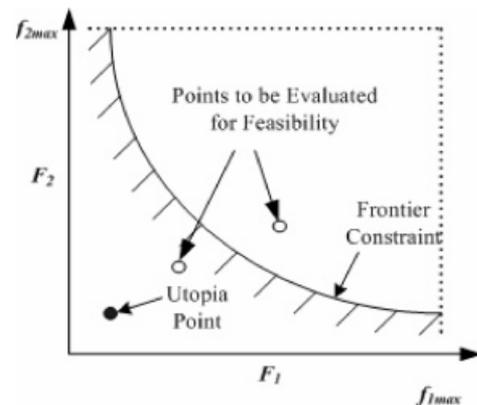


Fig. 10. The identification of potential points for feasibility testing.

tion for fitting the Pareto points. The generalized equation for this model is shown in Eq. (7).

$$a_0 + \sum_{i=1}^{n-1} a_i f_i + \sum_{i=1}^{n-1} b_i f_i^2 = f_n, \quad (7)$$

in which  $f_i$  represents the  $i$ th objective in our multi-objective performance space. The coefficients  $a_0$ ,  $a_i$ , and  $b_i$  are determined using the method of least squares with no side constraints on them.

- b. Constrained quadratic function: with no side constraints in place, the representation of the frontier constraint developed using Eq. (7) may not necessarily exhibit the inverse relationships between objectives inherent in the definition of Pareto optimality. The Pareto function could also “fold over” upon itself within the bounds of the objectives, causing some feasible design points to lie below the representation of the Pareto frontier in the infeasible region. To prevent this from occurring, the coefficients in Eq. (7) are determined using the method of least squares but with additional side constraints imposed on the coefficient values. If the coefficients are constrained such that the surface always has a nonpositive gradient, it will necessarily exhibit inverse relationships between objectives and the folding over of the surface will also be prevented. The generalized partial gradient for the  $i$ th objective is given in Eq. (8).

$$\frac{\partial f_n}{\partial f_i} = a_i + 2b_i f_i. \quad (8)$$

When all objectives are being minimized, the slope of the Pareto set is always negative. Therefore, the partial derivatives of Eq. (8) need to be negative, as shown in Eq. (9).

$$\frac{\partial f_n}{\partial f_i} = a_i + 2b_i f_i \leq 0. \quad (9)$$

Further, let  $f_i$  be normalized between 0 and 1, with 0 being the most desired value and 1 being the least desired objective value (because minimization of all objectives is sought). The largest (worst case) value that  $\partial f_n / \partial f_i$  can take is  $a_i + 2b_i$ . Hence, it is sufficient to ensure that these worst case values are less than or equal to zero.

Therefore, this model may be expressed mathematically as given in Eq. (10).

$$a_0 + \sum_{i=1}^{n-1} a_i f_i + \sum_{i=1}^{n-1} b_i f_i^2 = f_n, \quad (10)$$

$$a_i + 2b_i \leq 0 \quad i = 1, \dots, n.$$

Once again, the method of least squares is used to determine the value of the coefficients. Of the two surface-fitting models described in this section, the one with the higher  $R^2$  value is used to represent the Pareto set.

Before discussing how the mathematical representation of the Pareto set is used for feasibility assessment, results of surface fitting for the three-objective example problem are presented.

Using the Pareto set generated by the MOGA, the two methods presented in this section are applied to determine mathematical expressions of the Pareto surface for the three-objective problem. For both cases, a least-squares regression was applied to the data to develop the coefficients of the polynomial surface in MATLAB. For the constrained surface, the optimization toolbox within MATLAB was also used to determine the coefficients while handling the constraints placed upon the model. The  $R^2$  values for the two different models are given in Table 5.

As seen in Table 5, the unconstrained quadratic model exhibits substantially less fitting error than the constrained quadratic model. This result is as expected due to the constraints imposed on the allowable values of the coefficients. All Pareto sets are required to have nonpositive gradients because the objective functions forming the Pareto frontier are assumed to be minimized, and are competing against each other. Because the Pareto set for this problem has a non positive gradient as seen in Figure 6, and the constrained model provides a low fit, the unconstrained model is used for this example. In addition, for problems with Pareto sets known to having nonpositive gradients, using a constrained model would be unnecessary.

The coefficients of the unconstrained surface of Eq. (7) are given in Eq. (11).

$$a_0 = 1.47922, \quad b_1 = 12.36493,$$

$$a_1 = -2.87069, \quad b_2 = 2.52139,$$

$$a_2 = -2.96181. \quad (11)$$

Thus, a mathematical representation of the Pareto frontier is obtained. The final step of the TFM is to use this surface for feasibility assessment. This is discussed in the next section.

**Table 5.**  $R^2$  values for Pareto surface models

| Pareto Surface Model              | $R^2$ |
|-----------------------------------|-------|
| Unconstrained quadratic [Eq. (7)] | 0.891 |
| Constrained quadratic [Eq. (10)]  | 0.376 |

### 3.4. Feasibility assessment tool

The feasibility of a proposed vehicle is determined based on the geometric location of the test point relative to the Pareto frontier constraint and utopia point. The steps of the method, termed the *ray method*, are as follows:

1. The equation of the line joining the utopia point and the candidate test point is generated. The vector form of an  $n$ -dimensional equation is given in Eq. (12).

$$\bar{r} = \bar{r}_0 + t\bar{\eta}, \quad (12)$$

where  $\bar{r}$  is the  $N$ -dimensional variable vector,  $\bar{r}_0$  is the  $N$ -dimensional point (corresponding to either the utopia point or the test point),  $\bar{\eta}$  is the slope vector (determined using the utopia and test point), and  $t$  is the variable parameter (representing a single degree of freedom).

Consider a design point ( $P$ ), which is represented as

$$P = (f_{10}, f_{20}, f_{30}, \dots, f_{n0}),$$

and the utopia point ( $U$ ) of the performance space (assuming minimization of all objectives is desired), which is given as

$$U = (f_{1\min}, f_{2\min}, f_{3\min}, \dots, f_{n\min}).$$

Then, the parametric form of Eq. (12) is given as Eq. (13):

$$\begin{aligned} f_1 &= f_{10} + t(f_{1\min} - f_{10}), \\ f_2 &= f_{20} + t(f_{2\min} - f_{20}), \\ f_3 &= f_{30} + t(f_{3\min} - f_{30}). \end{aligned} \quad (13)$$

Equation (13) is essentially the straight line equation in  $n$  dimensions, with  $t$  representing the single degree of freedom of a straight line.

2. Next, the point of intersection of the generated line and the Pareto frontier constraint is determined. To do so, Eq. (13) is substituted into either Eqs. (7) or (10) (depending on the model used). This yields one equation with the single unknown  $t$ .
3. The equation determined in step 2 is solved for  $t$ . The determined value of  $t$  is substituted into Eq. (13), producing a single solution that is the point of intersection of the line joining the utopia point and the test point with the Pareto frontier constraint [Eqs. (7) or (10)].
4. Finally, the distances from the utopia point to the intersection point and from the utopia point to the test

point are evaluated. There are three possible cases that could occur depending upon this distance.

- If the distance from the utopia point to the point of intersection is *greater than* the distance from the utopia point to the test point, then the test point is infeasible (below the Pareto frontier constraint). This is shown graphically using a 2-D example in Figure 11.
- If the distance from the utopia point to the point of intersection is *less than* the distance from the utopia point to the test point, then the test point is feasible and dominated by the Pareto set (above the Pareto frontier constraint). This implies that the test point is indeed feasible, but technically not as superior as it could be. This is shown for a 2-D example in Figure 12.
- If the distance from the utopia point to the point of intersection is *equal to* the distance from the utopia point to the test point, then the test point is Pareto optimal. This implies that the test point is at the current limit of performance and is a potentially superior design.

The developed feasibility assessment methodology is illustrated for the three-objective problem next. In this paper the process of evaluating feasibility is illustrated using performance values determined from the nature of the performance space. Because the entire performance space is known, data points from the regions that are known to be feasible and infeasible are generated and used in the code developed to determine feasibility. Note that when generating this data, only the objective function values are provided with no design variable information. Determining the corresponding design variable values once a set of performance measures are determined to be feasible is an area of ongoing work and the methods developed for this are presented in Ferguson et al. (2005). Using this data set serves two purposes: not only does it illustrate the feasibility assess-

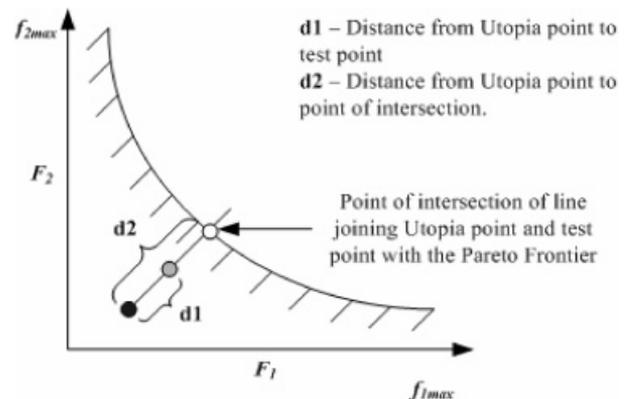


Fig. 11. Infeasible point identification using the ray method.

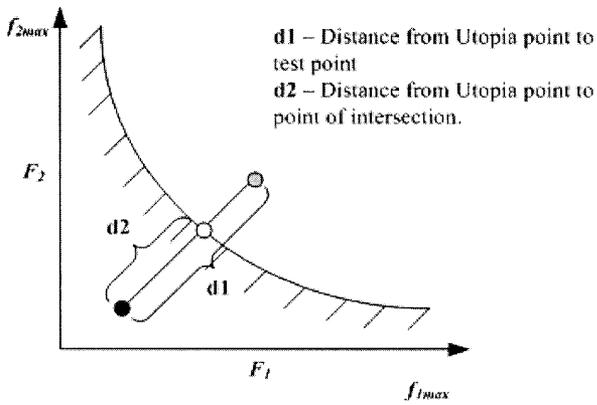


Fig. 12. Feasible point identification using the ray method.

ment process, it also provides insight into the validity of the model. The results of applying the ray method to test for feasibility for the unconstrained function representation of the Pareto frontier for the case study problem are shown in Table 6.

Looking closely at the results shown in Table 6, test points 1 and 2 lie above the Pareto set of points in the performance space resulting in  $t$  values less than 1, thus entitling them *feasible*. The third test point is *out of bounds*, because the performance measure values are better than the utopia point and the  $t$  value is negative, implying no intersection. Test point 4 is said to be *feasible*, although based on the distance measures, the point lies precisely on the Pareto set. This is also validated from the  $t$  value of 1. The computational implementation of this method calls this point feasible as opposed to Pareto optimal due to numerical error. However, because the feasibility assessment tool is used to only determine if a point is feasible from the engineering domain, calling test point 4 feasible is sufficient. Finally, test point 5 is said to be infeasible because it lies below the Pareto set of points and has a  $t$  value greater than 1. The performance values shown in Table 6 are plotted along with the Pareto set of points and are shown in Figure 13. Thus, it is seen that the feasibility assessment tool behaves exactly as it is expected to for the three-objective case study problem.

In concluding this section, it is seen that in three dimensions, the application of the feasibility test is trivial. Deal-

ing with a three-dimensional performance space allows for easy visualization and the feasibility of a test point can clearly be seen. However, when dealing with problems of greater than three objectives, using visualization techniques no longer becomes a simple task. Application of the ray method allows for quick and efficient testing of numerous combinations of desired objective values with very little computational expense. To demonstrate this for the GM problem, feasibility is tested using specifications from 78 late-model sedans. This problem consists of five-objective functions, 10 design variables, and three constraints. Because of the large number of dimensions in this problem, the surface that is created is not only difficult to understand, but almost impossible to visually use when determining the feasibility of a given test point. The next section describes the application of the TFM to the GM problem, which was the original motivation for this work.

#### 4. RESULTS FOR GM PROBLEM

In this section, the developed TFM is applied to the GM problem. Section 4.1 formulates the problem, Section 4.2 provides results from the gap analyzer, Section 4.3 describes the results from fitting a surface to the Pareto set of points, and finally, Section 4.4 provides the results of feasibility assessment of 78 benchmark cars determined using the developed Pareto frontier constraint.

##### 4.1. Problem formulation

For this study, a multidisciplinary vehicle design framework developed within the GM R&D Center is used to determine the values of the objectives for a vehicle design given a vector of design variables. The design variables used were a combination of 10 high-level vehicle geometric parameters and discrete design configuration choices. The objectives were five vehicle attributes from several engineering disciplines such as powertrain performance and occupant packaging. The Pareto optimal data points are obtained from the MOGA. This system of 10 design variables and five objectives forms the black box analysis routine. Although the details of this routine are not paramount to this paper, more technical details can be found in Bennett et al. (1995). In the next section, the gap analyzer methodology introduced in Section 3.2 is used to study the gaps in this example problem.

##### 4.2. Gap analyzer results

The final population produced by the MOGA initially applied to a system does not necessarily guarantee a complete Pareto set. As discussed in Section 3.2, there exist the possibilities of gaps, or discontinuities, within the frontier that could affect the fidelity of the representation of the Pareto frontier. This can be seen in the 2-D projection of the Pareto set generated in this project shown in Figure 14.

Table 6. Feasibility results for test points

| Number | F1     | F2     | F3     | $t$    | Feasibility Results |
|--------|--------|--------|--------|--------|---------------------|
| 1      | 4.0    | 5.0    | 4.0    | 0.2759 | Feasible            |
| 2      | 3.0    | 3.0    | 3.0    | 0.4704 | Feasible            |
| 3      | 0.0    | 0.0    | 0.0    | -10.68 | Out of bounds       |
| 4      | 1.1059 | 2.1044 | 2.5518 | 1.0    | Feasible            |
| 5      | 1.0    | 2.0    | 2.2    | 1.2994 | Infeasible          |

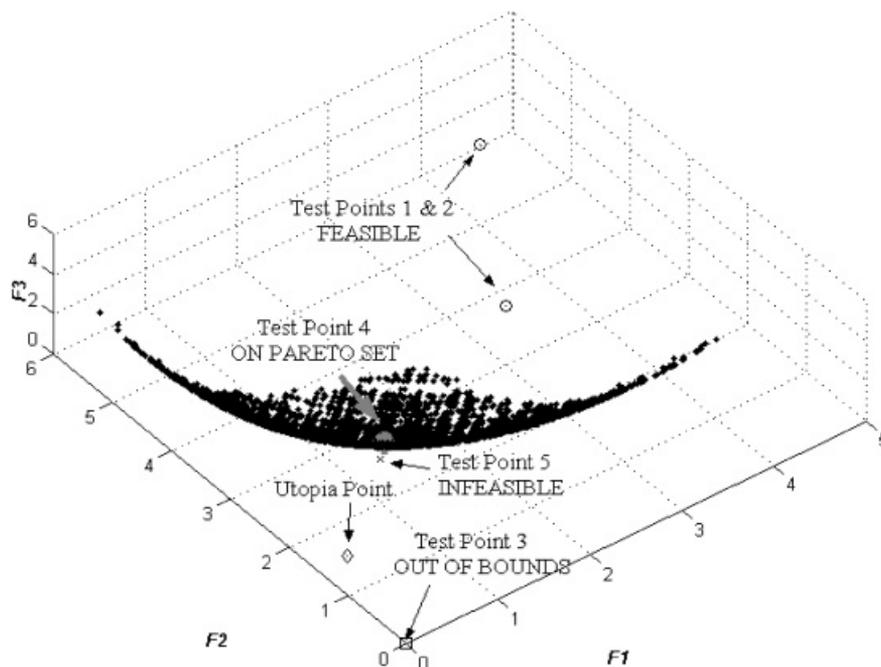


Fig. 13. The feasibility of the test points in the performance space.

From Figure 14, it can be seen that for two of the objectives the Pareto frontier is not a smooth curve, but instead is composed of clusters of Pareto optimal points. The objectives are referred to as *objective A* and *objective B*, because the actual objective names are confidential information. Analysis of these points with the gap analyzer developed in this work identifies the regions between the clusters as potential gaps in the Pareto frontier. However, the objectives plot-

ted in Figure 14 are functions of the type of powertrain, which is a discrete design variable. The discrete design variable results in a Pareto surface that is discontinuous. That is, the Pareto surface is comprised of both clustered and nonclustered regions (each cluster corresponds to an individual powertrain type). The gap analyzer operates in the performance space and correctly identifies these nonpopulated regions as gaps. In addition, the gap analyzer does not

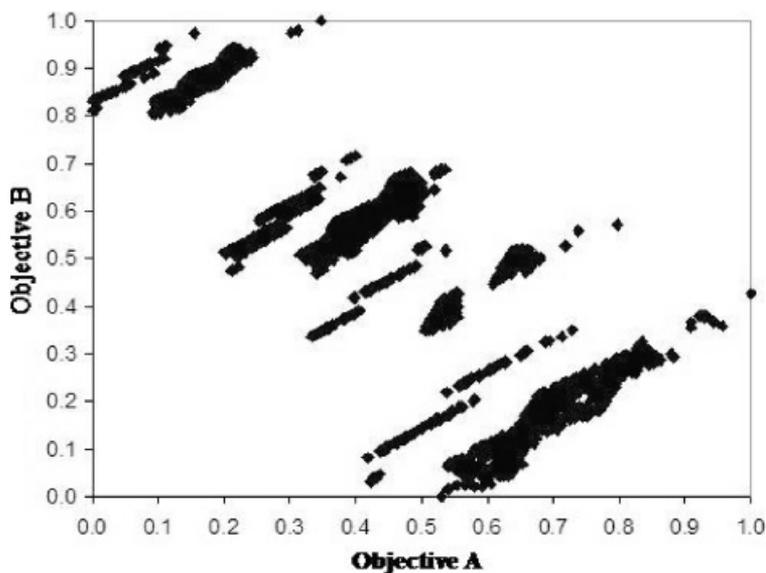


Fig. 14. The results of the gap analysis study for the GM problem.

find any gaps within the clustered regions of the Pareto frontier. This is because an exhaustive MOGA is used for populating the Pareto surface. Because there cannot exist any points between the clusters (due to the discrete design variable) and there exist no gaps within the clustered regions, it is concluded that no gaps that can be filled exist. In the next section, it is assumed that a continuous curve can be used to fit the Pareto points. It is acknowledged that this assumption might not be valid at all times because it is already known that the Pareto set is not a continuous curve. However, developing a surface representation of a mixed discrete-continuous data set is outside the scope of this work, and is a topic for future research. Section 4.3 presents the results of fitting a continuous curve to the Pareto set of points.

### 4.3. Surface fitting results

Using the Pareto set generated by the MOGA, the two methods presented in Section 3.3 are applied to determine mathematical expressions of the Pareto surface. The unconstrained quadratic model with  $R^2 = 0.842$  exhibits substantially less fitting error than the constrained quadratic model with  $R^2 = 0.672$ . This result is as expected because of the constraints imposed on the allowable values of the coefficients. The coefficients corresponding to the  $f_i$  term, namely  $a_j$  are all

nonpositive based on the second constraint. In this application, the side constraints applied to the constrained quadratic model are active for two of the five linear terms included in the model. It is noteworthy that although the constrained model has a lower  $R^2$  value, it represents the strictly nonpositive gradient region of the Pareto surface.

Having  $R^2$  values from both surfaces, the next step is to examine the results for residuals from the Pareto frontier fitting process. This examination is performed for the constrained quadratic model. These results are obtained using the optimization toolbox in MATLAB. Optimization is used to ensure the feasibility of the coefficient values with respect to the constraints specified in Eq. (10).

As shown in Figure 15, it appears that there are at least two distinct, discrete sections of the Pareto frontier represented by the quadratic response surface. The largest section (section with the most points) appears to have a normal distribution centered on zero. There is also another small section with a mean less than zero.

These results support our previous observation from application of the gap analyzer that there do indeed seem to be gaps in the Pareto frontier in which no feasible points exist. As discussed earlier, these sections seem to be a function of the discrete design configuration variables included in this problem. When different design configuration settings are selected, it appears to introduce breaks in the Pareto sur-

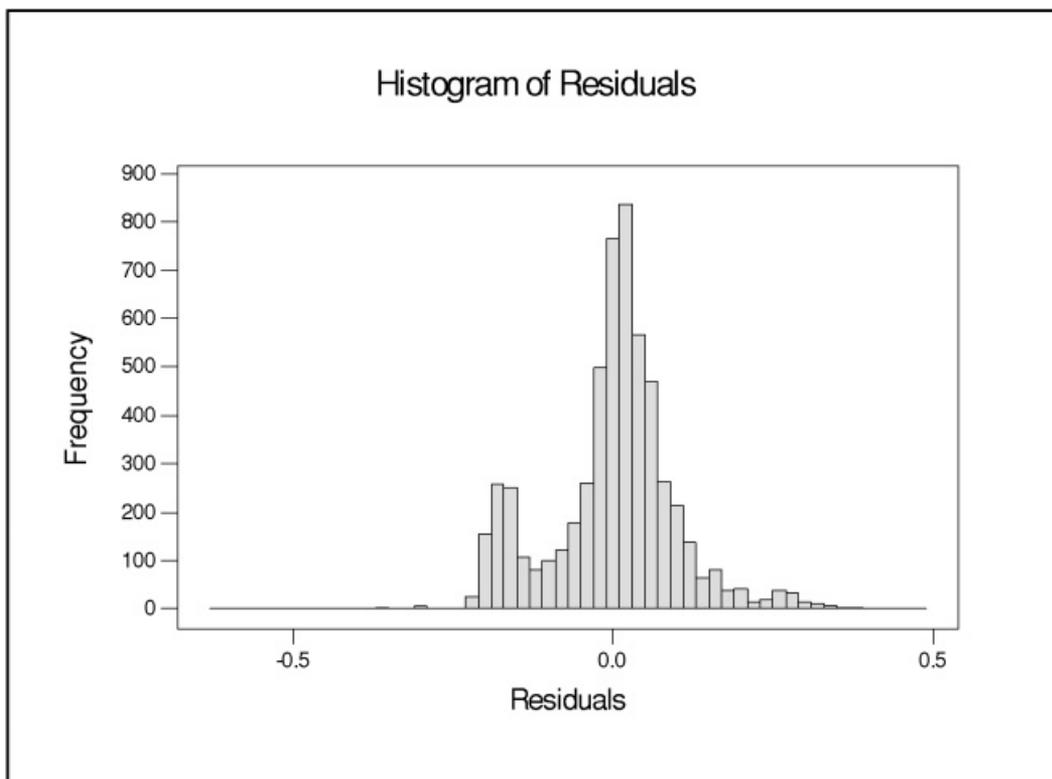


Fig. 15. A histogram of the constrained residuals.

face. Investigation of the relative merits of fitting separate response surfaces to each distinct section of the Pareto frontier versus representing the entire performance space with a single response surface is an area of future research. In the next section, the feasibility assessment tool described in Section 3.4 is used to determine the feasibility of some existing passenger vehicles against the two Pareto surfaces generated in this section.

#### 4.4. Feasibility assessment results

Once the expressions for the Pareto surfaces are generated, they may be used to evaluate new sets of objective functions for feasibility. Typically, these sets would be desired combinations of objectives specified by marketing. However, in this paper the process of evaluating feasibility is illustrated using specifications from 78 late-model sedans. Using data from production vehicles serves two purposes: not only does it illustrate the feasibility assessment process, it also provides insight into the validity of the model. When evaluating data from production vehicles, the expected results would be that some vehicles fall well within the feasible region, some vehicles (most likely the recently introduced vehicles) will lie very close to the Pareto frontier constraint, and no vehicles will fall well within the infeasible region. Observation of this type of result suggests that the Pareto frontier constraint truly represents the efficient frontier of vehicle design. On the other hand, if production vehicles are observed to be within the infeasible region or if all of the production vehicles are observed to be well within the feasible region, then it is very likely that the Pareto frontier constraint generated using the black box system does not truly represent the efficient frontier for vehicle design, and adjustments should therefore be made to better correlate the analytical results with the observations of production vehicles.

The results of applying the ray method to test for feasibility for the two representations of the Pareto frontier are shown in Table 7. Of the vehicles failing the feasibility test, 17 of them failed because they have a better value in at least one objective than the best value found in the Pareto set. In each case, this was because the vehicles employed technologies that were beyond the scope of the multidisciplinary analysis system used to generate this Pareto surface. This is

not only an expected result, but also an encouraging one because it demonstrates the consistency of the TFM: vehicles employing technologies within the scope of the underlying black box analysis system pass the feasibility test as expected whereas those employing technologies outside the bounds of the underlying analysis system fail the feasibility test, which was also expected.

The other infeasible vehicles (two for the unconstrained surface and one for the constrained surface) were found because they each employed a 1.8-L engine, which was not modeled in the black box vehicle model. The smallest engine modeled in the analysis model was a 2.2-L engine. These cars, therefore, have a better fuel economy than what the model could predict but comparable values for the other objectives, and hence, end up below the frontier constraint. This information can be used by engineers to update the vehicle model by including smaller engines in order to get a more accurate representation of the bounds of current vehicle performance.

The difference in the number of feasible cars using the different surfaces is due to fidelity of the surface fit. Vehicles very close to the frontier could be deemed feasible using one method and infeasible using another simply based on precisely how the regression surface passes through the data points.

Thus, in this section, we determined a mathematical representation of the Pareto surface, used it as a constraint to assess the feasibility of 78 existing vehicle designs, and analyzed the reasons for infeasibility of the benchmark vehicles. In the next section, we provide final conclusions to this paper and cite some avenues for future work.

## 5. CONCLUSIONS AND FUTURE WORK

In this work, we have successfully developed an approach to determine whether or not a given set of objective functions, actual or hypothetical, is feasible with respect to a Pareto frontier obtained from a multiobjective analytical vehicle model. The Pareto frontier serves as a constraint in the determination of the *engineering* feasibility of these desired performance measures. This method can be used to effectively map the performance limits for a set of available technologies and to integrate gap analysis, meta-modeling, and feasibility assessment algorithms to assess the feasibility of a vector of vehicle objective functions. By running a coarse MOGA on the entire problem and using the gap analyzer, a computationally efficient sampling of the performance space can result in a high fidelity representation of the Pareto set. Application of the approach using benchmarking data from 78 late-model sedans indicates that at least 17 of them are not feasible in at least one performance attribute given the vehicle technologies considered in this project. These methods and tools provide vehicle engineers with critical information that allows them to define product specifications while maintaining

**Table 7.** Feasibility results of GM problem

|                       | No. of Feasible Vehicles |
|-----------------------|--------------------------|
| Unconstrained surface | 59 of 78                 |
| Constrained surface   | 60 of 78                 |

high confidence for successful realization of the vehicle's design.

In this paper, the TFM is built around a Pareto frontier obtained from the current technology developed. In addition, the Pareto set is unique to only a family of products. For example, in this paper, the sedan class of automobiles is only being considered. An important issue is related to determining feasibility with new technological developments, because the Pareto frontier would change as new objectives or new bounds to current objectives are considered in the product design process. As part of future work, it is hypothesized that for a single class of products, and for minor technological advancements where the objectives remain the same, a "fuzzy" Pareto set can be utilized. A fuzzy Pareto set is one where the current frontier is shifted toward improvement of the improved objectives by shifting the bounds and curvature of the frontier. For major technical advancements, however, an entirely new frontier, and hence TFM would need to be developed. For example, a minor technological advancement could be an improvement in engine performance, thereby improving fuel economy. This advancement can be easily incorporated into the current TFM by shifting the Pareto frontier. However, a major technology improvement such as hydrogen fuel cells makes the notion of fuel economy as a design objective totally irrelevant. For such an automobile, an entirely new TFM would need to be set up.

Future work in this area also includes expanding the capabilities of the analytical model to include additional vehicle attributes as well as the additional degrees of design freedom required to support them. This expansion may require the application of parallel processing techniques and the development of a more sophisticated MOGA to reduce the time necessary for sampling of the design space. As the scope of the performance space expands, the gap analyzer will play an increasingly important role in maintaining a high level of efficiency when populating the Pareto set by reducing the overall number of evaluations and generations required by the MOGA. Additional information about the relationships between performance space and design space in the neighborhood of a given test point could also provide additional benefit when considering trade-offs in this multi-objective preliminary design process.

## ACKNOWLEDGMENTS

We thank the GM Corporation and the National Science Foundation (Grant DMII-9875706) for their support of this research.

## REFERENCES

- Alzubbi, A., Ndiaye, A., Mahdavi, B., Guibault, F., Ozell, B., & Trepanier, J.Y. (2000). On the use of JAVA and RMI in the development of a computer framework for MDO. *8th AIAA Symp. Multidisciplinary Analysis and Optimization*, Paper No. AIAA-2000-4903, Long Beach, CA.
- Azarm, S., Reynolds, B.J., & Narayanan, S. (1999). Comparison of two multi-objective optimization techniques with and within genetic algorithms. *ASME Int. Design Engineering Technical Conf.*, Paper No. DETC99/DAC-8584.
- Balling, R.J. (2000). Pareto sets in decision-based design. *Engineering Valuation and Cost Analysis* 3, 189–198.
- Bennett, J.A., Botkin, M.E., Koromilas, C., Lust, R.V., Neal, M.O., Wang, J.T., & Zwiers, R.I.A. (1995). Multidisciplinary framework for preliminary vehicle analysis and design. *Proc. ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization*, pp. 3–21, SIAM, Hampton, VA.
- Coello, C.A.C., & Aguirre, A.H. (2002). Design of combinational logic circuits through an evolutionary multiobjective optimization approach. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16(1), 39–53.
- Das, I., & 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(1), 63–69.
- Davis, L. (1991). *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold.
- Deb, K., & Kumar, A. (1995). Real-coded genetic algorithms with simulated binary crossover: studies on multimodal and multiobjective problems. *Complex Systems* 9, 431–454.
- Eddy, J., & Lewis, K. (2001). Effective generation of pareto sets using genetic programming. *ASME Int. Design Engineering Technical Conf.*, Paper No. DETC01/DAC-21094, Pittsburgh, PA.
- Eldred, M.S., Swiler, L.P., Gay, D.M., Brown, S.L., Giunta, A.A., Wojtkiewicz, S.F., Hart, W.E., & Watson, J.P. (2004). *DAKOTA: A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis, Version 3.2 Reference Manual*, Paper No. SAND2001-3515. Albuquerque, NM: Sandia National Laboratories.
- Fenyes, P.A., Donndelinger, J.A., & Bourassa, J.F. (2002). A new system for multidisciplinary design and optimization of vehicle architectures. *9th AIAA Symp. Multidisciplinary Analysis and Optimization*, Paper No. AIAA-2002-5509, Atlanta, GA.
- Ferguson, S., Gurnani, A., Donndelinger, J., & Lewis, K. (2005). A study of convergence and mapping in preliminary vehicle design. *International Journal of Vehicle Systems Modeling and Testing* 1(1–3), 192–215.
- Fonseca, C.M., & Fleming, P.J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* 3(1), 1–16.
- Fonseca, C.M., & Fleming, P.J. (1998). Multi-objective optimization and multiple constraint handling with evolutionary algorithms—part 1: a unified formulation. *IEEE Transactions on Systems, Management, and Cybernetics—Part A: Systems and Humans* 28(1), 26–37.
- Fu, Z., & de Pennington, A. (1993). Constraint based design using an operational approach. *Research in Engineering Design* 5(3/4), 202–217.
- Goel, T., Vaidyanathan, R., Haftka, R.T., Queipo, N.V., Shyy, W., & Tucker, K. (2004). Response surface approximation of Pareto optimal front in multi-objective optimization. *10th AIAA/ISSMO Symp. Multidisciplinary Analysis and Optimization*, Paper No. AIAA-2004-4501, Albany, NY.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.
- Gurnani, A., & Lewis, K. (2005). Robust multiattribute decision making under risk and uncertainty in engineering design. *Engineering Optimization* 37(8), 813–830.
- Kasprzak, E., & Lewis, K. (2000). An approach to facilitate decision trade-offs in Pareto solution sets. *Journal of Engineering Valuation and Cost Analysis* 3(1), 173–187.
- Kosaka, I., Charpentier, C., & Watson, B. (2000). An interface between SDRG I-DEAS and the genesis structural analysis and optimization code. *8th AIAA Symp. Multidisciplinary Analysis and Optimization*, Paper No. AIAA-2000-4933, Long Beach, CA.
- Koski, J. (1985). Defectiveness of weighting method in multicriterion optimization of structures. *Communications in Applied Numerical Methods* 1(6), 333–337.
- Koza, J., Keane, M., Streeter, M., Adams, T., & Jones, L. (2004). Invention and creativity in automated design by means of genetic programming. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 18(3), 245–269.
- Kurapati, A., Azarm, S., & Wu, J. (2002). Constraint handling improvements for multi-objective genetic algorithms. *Structural and Multidisciplinary Optimization* 23(3), 204–213.

- Lee, J., & Hajela, P. (1996). Parallel genetic algorithm implementation in multidisciplinary rotor blade design. *Journal of Aircraft* 33(5), 962–969.
- Lee, J., Li, D.J., Liu, X., Soderburg, N., Sudjianto, A., Vora, M., & Wang, S. (2001). An approach to robust design employing computer experiments. *ASME Int. Design Engineering Technical Conf.*, Paper No. DETC01/DAC-21094, Pittsburgh, PA.
- Longacre, K., Vance, J.M., & DeVries, R.A. (1996). Computer tool to facilitate cross-attribute optimization. *6th AIAA Symp. Multidisciplinary Analysis and Optimization*, Paper No. AIAA-96-4132-CP, Bellevue, WA.
- Marinescu, R., & Dechter, R. (2005). Advances in AND/OR branch-and-bound search for constraint optimization. *7th Int. Workshop on Preferences and Soft Constraints of the Eleventh Conf. Principles and Practice of Constraint Programming*.
- Messac, A., & Sundararaj, J.G. (2000). Physical programming's ability to generate a well-distributed set of pareto points. *41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conf.*, AIAA 2000-1666.
- Messac, A., Sundararaj, J.G., Tappeta, R.V., & Renaud, J.E. (2000). The ability of objective functions to generate points on non-convex Pareto frontiers. *AIAA Journal* 38, 1084–1091.
- Miller, B.L., & Goldberg, D.E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation* 4(2), 113–131.
- Narayanan, S., & Azarm, S. (1999). A multi-objective interactive sequential hybrid optimization technique for design decision making. *Engineering Optimization* 32, 485–500.
- Pareto, V. (1896). *Cours D'Économie Politique*, Vols. I and II. Lausanne, Switzerland: Rouge.
- Pareto, V. (1906/1971). *Manuale di Economica Politica* [Manual of Political Economy] (Schwier, A.S., Trans.). Milan, Italy/New York: Società Editrice Libraià/Macmillan.
- Port, O. (1998). Making complexity look simple: Ford, DuPont, and Tyco are among the fans of breakthrough design software. *Business Week* May 18, 166.
- Röhl, P.J., Kolonay, R.M., Irani, R.K., Sobolewski, M., Kao, K., & Bailey, M.W. (2000). A federated intelligent product environment. *8th AIAA Symp. Multidisciplinary Analysis and Optimization*, Paper No. AIAA-2000-4902, Long Beach, CA.
- Schmidt, L., Shi, H., & Kerkar, S. (2005). A constraint satisfaction problem approach linking function and grammar-based design generation to assembly. *Journal of Mechanical Design* 127(2), 196–205.
- See, T.-K., Gurnani, A., & Lewis, K. (2004). Multiattribute decision making using hypothetical equivalents and inequivalents. *Journal of Mechanical Design* 126(6), 950–958.
- Siddique, Z., & Boddu, K.R. (2004). A mass customization information framework for integration of customer in configuration/design of a customized product. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 18(1), 71–85.
- Stender, J., Ed. (1993). Parallel genetic algorithm: theory and applications. In *Frontiers in Artificial Intelligence and Applications*, Vol. 14. Amsterdam: IOS Press.
- Townsend, J.C., Samareh, J.A., Weston, R.P., & Zorumski, W.E. (1998). *Integration of a CAD System Into an MDO Framework*, Technical Report NASA/TM-1998-207672. Langley, CA: NASA Langley Research Center.
- Viennet, R., Fontiex, C., & Marc, I. (1996). Multiobjective optimization using a genetic algorithm for determining a Pareto set. *Journal of Systems Science* 27(2), 255–260.

---

**Kemper Lewis** is a Professor in the Department of Mechanical and Aerospace Engineering at the University at Buffalo–SUNY, Buffalo, NY. He is also Executive Director of the New York State Center for Engineering Design and Industrial Innovation. His research areas include design theory, decision-based design, distributed design, reconfigurable systems, multidisciplinary optimization, and multiobjective design. He has over 100 technical publications in these research areas. His awards include the Society of Automotive Engineers Teetor Award, the ASME Black & Decker Best Paper Award, the State University of New York Chancellor's Award for Excellence in Teaching, and the National Science Foundation Career Award.

**Joseph Donndelinger** is a Staff Research Engineer in the Vehicle Development Research Laboratory at the General Motors Research & Development Center in Warren, MI. He earned his MS degree in industrial engineering and his BS degree in mechanical engineering, both from the University of Illinois at Urbana–Champaign. His 10 years of experience with General Motors and the Ford Motor Company have broadly spanned the vehicle development process, with assignments in subsystem design, vehicle integration, advanced technology development, and process reengineering. His research interests include optimization, decision making, and methods for design conceptualization, decomposition, and synthesis.

**Ashwin Gurnani** received his BS (2002) and MS (2003) in mechanical engineering from the University at Buffalo–SUNY, Buffalo, NY. He is currently working on his PhD, and his dissertation focuses on decentralized design and the study of rationality in design decisions.

**Scott Ferguson** received his BS (2002) and MS (2004) in mechanical engineering from the University at Buffalo–SUNY, Buffalo, NY. He is currently working toward his PhD, and his dissertation focuses on the design and application of reconfigurable systems.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.