AIAA JOURNAL
Vol. 41, No. 6, June 2003

Concept Selection Using s-Pareto Frontiers

Christopher A. Mattson* and Achille Messac†
Rensselaer Polytechnic Institute, Troy, New York 12180

We introduce the notion of s-Pareto optimality and show how it can be used to improve concept selection in engineering design. Specific design alternatives are classified as s-Pareto optimal when there are no other alternatives from the same or any other general design concept that exhibit improvement in all design objectives. Further, we say that the set of s-Pareto design alternatives comprises the s-Pareto frontier. Under the proposed approach the s-Pareto frontier plays a paramount role in the concept selection process, as it is used to define and classify concept dominance. The s-Pareto frontier-based concept selection method can be characterized as one that capitalizes on the benefits of computational optimization during the conceptual phase of design, before a general design concept has been chosen. An introduction to s-Pareto optimality and a method for generating s-Pareto frontiers are developed. An approach for using s-Pareto frontiers to perform concept selection is also presented. The methods proposed can effectively aid in the elimination of dominated design concepts, keep competitive concepts, and ultimately choose a specific design alternative from the selected design concept. A truss design problem is used to illustrate the usefulness of the method.

Nomenclature

g	=	vector of inequality constraints
h	=	vector of equality constraints
J	=	aggregate objective function
m_i	=	number of points along N_i
N_i	=	ith vector defining the utopia plane
n	=	number of design objectives
n_x	=	number of design variables
T^k	=	relaxation/slack variable for concept k
X_P	=	generic point on the utopia plane
x	=	vector of design variables
δ	=	increment by which feasible space is reduced
μ	=	vector of design objectives (or design metrics)
μ^{i*}	=	ith anchor point
μ^{*k}	=	optimum design objective value for concept k
μ^{si*}	=	s-anchor point for the ith objective

Subscripts and Superscripts

c	=	concept specific, for example, μ^{ck}
i, j, q	=	dummy indices
l	=	lower bound
s	=	for the set of concepts
u	=	upper bound
$*$	=	optimal

I. Introduction

ENGINEERING design can be divided into two major phases: conceptual design and detailed design. Conceptual design can be further divided into function specification, concept generation, and concept selection. Many in the design community accept the notion that more than 70% of the final product quality and cost are determined in the conceptual design phase, as early design decisions are made.[1–3] With this in mind it seems particularly prudent to make early design decisions in the most optimal manner possible. In this paper we focus on the conceptual design phase and attempt therein to capitalize on the benefits of computational optimization. More specifically, we focus on the important task of concept selection.

Before continuing, it is important to make a clear distinction between the way we use the terms *design concept* and *design alternative.* A design concept is an idea that has evolved to the point that there is a parametric model that represents the performance of the family of design alternatives that belong to that concept's definition. A design alternative is a specific design resulting from unique parameter values used in the parametric model of a concept. Design concepts are generated primarily during conceptual design, and a final design alternative is generally identified during detailed design. Other researchers have made similar distinctions between design concepts and design alternatives.[4,5]

Figure 1 illustrates the relationship between the impact of decision making and the typical level of decision-making rigor, at various stages in the design process. The motivation for this work is to increase design success by using more methodical and effective approaches for making decisions in conceptual design. Similar figures relating design knowledge and design freedom exist in the literature.[6]

With rare exceptions engineering design is a multiobjective activity that entails the resolution of conflicting design objectives.[7] Multiobjective optimization has proven to be an effective and powerful tool for resolving such conflicts in engineering design. An important class of solutions to the multiobjective problem is said to belong to the Pareto frontier. Each solution (or design alternative) comprising the frontier is understood to be Pareto optimal or dominant, which means there are no other design alternatives for which all objectives are better.[8–12] Edgeworth[13] was the first to identify this important class of solutions. Other early work in this area includes that of Pareto[14] and Koopmans.[8,15] For a given concept the selection of one Pareto design alternative over another is a matter of decision-maker preference. In this paper we consider the designer to be the primary decision maker and assume that he or she has considered the preference of all of the stakeholders.

Although Pareto optimality has played a significant role in the advancement of methods for multiobjective optimization, its use as a tool for conceptual design has not yet been fully explored. In this paper we present a method that uses principles of Pareto optimality to aid in selecting general design concepts, not specific design alternatives. The latter is the more traditional use of Pareto frontiers.[9,10] The method proposed herein can effectively aid to eliminate dominated (non-Pareto) design concepts, keep competitive concepts, and ultimately choose a specific design alternative from the chosen design concept.

Received 30 May 2002; revision received 10 January 2003; accepted for publication 20 January 2003.

*Ph.D. Candidate, Department of Mechanical, Aerospace, and Nuclear Engineering. Student Member AIAA.

†Associate Professor, JEC-2049, Department of Mechanical, Aerospace, and Nuclear Engineering; messac@rpi.edu. Member AIAA.

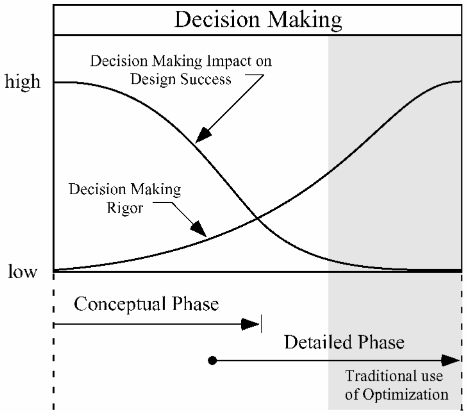


Fig. 1 Decision-making rigor in design.

There are various methods for concept selection.[16,17] Perhaps the most widely used methods in industry involve decision matrices.[17] Decision-matrix-based methods generally involve assigning a weight to each design objective, rating each design concept based on its estimated ability to meet a given design objective, and then performing the following summation:

$$S_k = \sum_{i=1}^{n} R_{ik} w_i \tag{1}$$

where S_k is the total score for concept k, w_i is the weight for the ith objective, and R_{ik} is the rating of concept k for the ith objective.

As Eq. (1) indicates, the mathematical structure of decision-matrix-based methods is a summation of weighted concept ratings. Optimization methods based on sums of weighted criteria have been outwardly criticized in the multiobjective optimization community.[18–21] The main limitation of weighted-sum methods is that they do not yield solutions that lie in nonconvex regions of the feasible design space. In practice, this means that weighted-sum methods, including decision-matrix-based methods, could miss potentially preferable design concepts.

Another popular method for concept selection is the Pugh method[22] (closely related to concept screening[17]). The Pugh method is a unique decision-matrix-based method that has the following distinctions. Criteria are not weighted (or they all have equal weights), and each concept is rated as "better than," "equal to," or "worse than" a reference concept. The Pugh method deliberately avoids the use of weights; Pugh himself argues that they are misleading in nature.[22] The Pugh method avoids some of the problems associated with other decision-matrix-based methods in that it forces design concepts to lie on convex points (boundaries) of the objective space.

Other research supports the notion of rigorous concept selection methods and includes the use of different numerical and optimization techniques. Patel et al.[23] use graph theory and linear physical programming to identify promising combinations of subsystems in multisystem design. Crossley et al.[5] use a genetic algorithm and combinatorial optimization for concept selection in conceptual aircraft design. Their methods are intended for use during a configuration definition phase, where types and number of aircraft engines (among other design features) are chosen. Perez et al.[24] also approach conceptual aircraft design using genetic algorithms. They conclude that genetic-algorithm-based methods are able to search a greater design space with more variable types and a wider variety of constraints than other optimization methods.

Other rigorous selection approaches for conceptual design include 1) topology optimization, which is often used to identify optimal geometries for the conceptual design of kinematic mechanisms and structural components[25]; 2) knowledge-based systems, which have been used to automate concept selection by drawing on results captured from previous designs[26]; and 3) fuzzy outranking models,[3] where linguistic terms (fuzzy numbers) are set by the designer and used to compare various design concepts. Concepts that are outranked (dominated) are removed, leaving only those that merit further development.

Popular methods such as decision matrices and the Pugh method lack the rigor that would otherwise bring improved structure, repeatability, and design exploration speed to the conceptual design process. Various new approaches have been used to bring added rigor to conceptual design. However, no method in the literature fully addresses the important multiobjective aspects of concept selection. That is, no method seeks to characterize the conceptual design space with Pareto frontiers, as we have done in this paper. Importantly, we will show that characterizing the design space with Pareto frontiers results in improved decision-making structure, repeatability, and speed of design space exploration.

The remainder of this paper is presented as follows. Section II presents conceptual preliminaries, including a brief overview of the normal constraint method for generating Pareto frontiers and the introduction of a Pareto filter. In Sec. III we 1) introduce the concept of s-Pareto optimality, 2) present formulations for obtaining the s-Pareto frontier, and 3) show how concept selection can be performed using an s-Pareto approach. In Sec. IV we present a truss design example, and in Sec. V concluding remarks are given.

II. Technical Preliminaries

This section presents requisite mathematical and conceptual preliminaries. In Sec. II.A a generic multiobjective optimization problem statement is given, which is followed by a formulation for obtaining the Pareto frontier endpoints. In Sec. II.B we present a terse overview of a new and effective method for generating Pareto frontiers, namely, the normal constraint[27] method. Later in this paper, a modified version of the normal constraint method is used to obtain the s-Pareto frontier. In Sec. II.C we introduce a Pareto filter that is designed to compare solutions and keep only those that are globally Pareto optimal.

A. Multiobjective Optimization Problem

Multiobjective optimization is a powerful means for resolving conflicting objectives in a computational setting. Mathematically, the multiobjective optimization problem can be stated by problem 1.

Problem 1 (P1): Multiobjective optimization problem statement

$$\min_x \{\mu_1(x), \mu_2(x), \ldots \mu_n(x)\} \qquad (n \geq 2) \tag{2}$$

Subject to:

$$g_q(x) \leq 0, \qquad q = 1, \ldots, r \tag{3}$$

$$h_j(x) = 0, \qquad j = 1, \ldots, v \tag{4}$$

$$x_{\mathrm{il}} \leq x_i \leq x_{\mathrm{iu}}, \qquad i = 1, \ldots, n_x \tag{5}$$

As P1 indicates, the multiobjective optimization problem does not yield a unique solution. To obtain a single optimum solution, the set of objectives in Eq. (2) is often replaced by a scalar function that is optimized. We call this function an aggregate objective function (AOF).

One multiobjective optimization approach is to objectively generate a set of optimal solutions to P1, followed by subjectively choosing the most attractive one. This approach has been referred to as generate-first–choose-later.[28] In this paper we use the generate-first–choose-later approach by first seeking to identify the Pareto optimal set, followed by subjectively choosing the most attractive design.

We identify the Pareto optimal set by generating the Pareto frontier. Figure 2a illustrates a feasible design region (shaded) and Pareto frontier (thick line) for a biobjective minimization problem. Three notable methods have proven effective in generating good representations of the Pareto frontier; they are the normal boundary intersection method,[20] physical programming,[28] and the normal constraint[27] method. A comparative study of these, and other, frontier generators is given in Messac et al.[29] In this paper we use the normal constraint method, which requires obtaining the Pareto frontier endpoints. We call these endpoints *anchor points* and obtain them using the following optimization problem statement.

Problem 2 (P2i): Obtaining anchor points

$$\min_x \mu_i(x), \qquad i = 1, \ldots, n \tag{6}$$

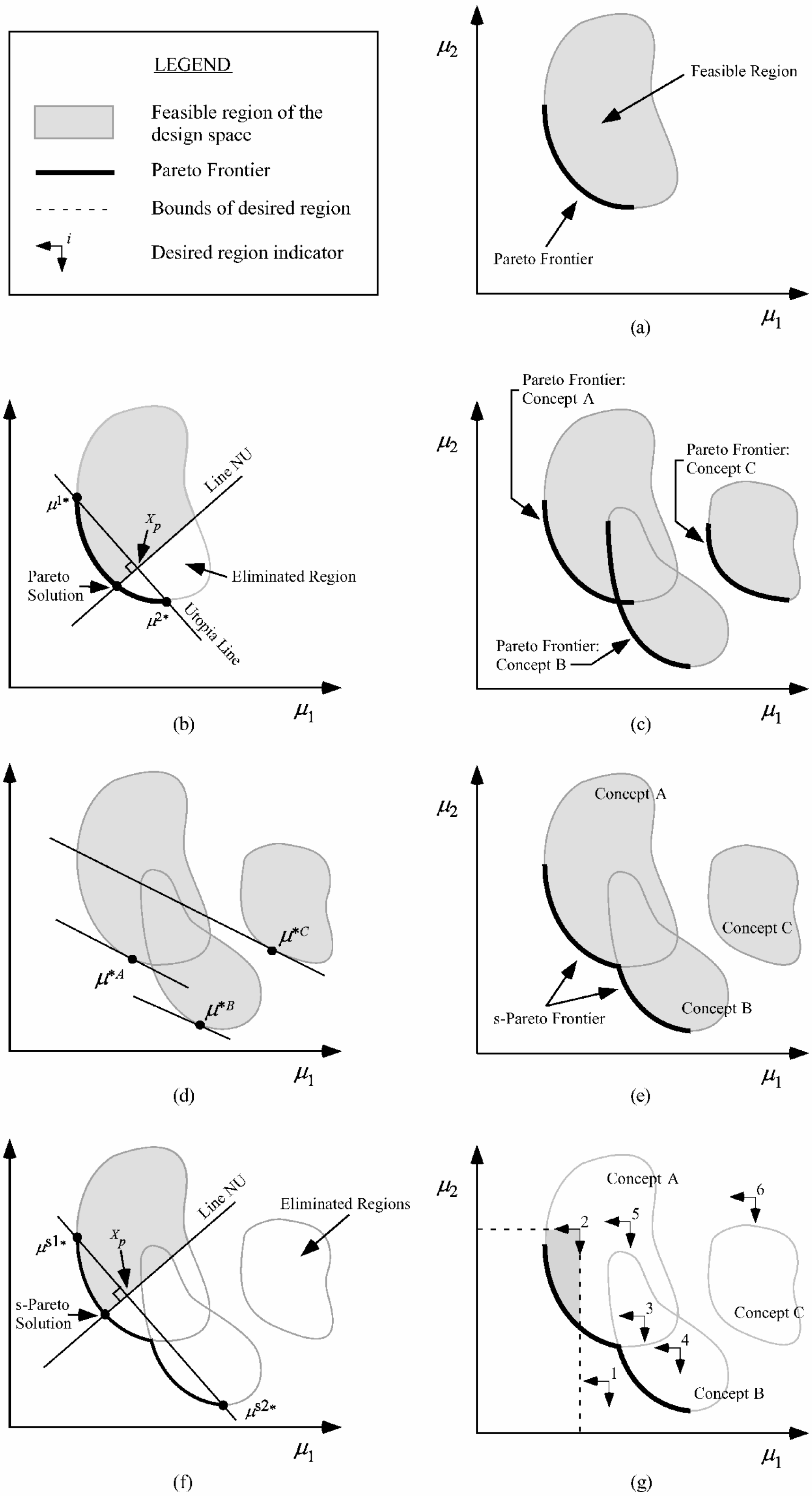


Fig. 2 s-Pareto frontier development and use.

subject to Eqs. (3–5). In the following section we present a brief overview of the normal constraint method, which is used later in our development.

B. Normal Constraint Method for Generating Pareto Frontiers

In this section we present a synopsis of the normal constraint method. A graphical description of the method for biobjective problems is given, followed by related optimization problem statements, which are given for generic problems of n objectives. For a more detailed description of the normal constraint method, the reader is referred to Ismail-Yahaya and Messac[27] and Messac et al.[29]

1. Graphical Description

The normal constraint method is a newly developed method for generating Pareto frontiers. Its basic objective is to efficiently generate a set of well-distributed Pareto solutions along the Pareto frontier. The normal constraint method converts a multiobjective optimization problem into a single-objective problem with added constraints.

These constraints are used to reduce the design space. After optimizing the single-objective problem subject to the constraints associated with the original problem and new added constraints, the result is a single Pareto solution for the multiobjective problem.

For simplicity of presentation, we graphically describe the normal constraint method for the biobjective minimization case depicted in Fig. 2a. The two design objectives to be minimized are denoted by μ_1 and μ_2, and the feasible design region is represented by the shaded area. The Pareto frontier associated with this feasible space is highlighted by the thick curve. The anchor points μ^{1*} and μ^{2*}, which are shown in Fig. 2b, are obtained by solving P2-1 and P2-2, respectively. A so-called utopia line is constructed between the two anchor points. Then, at a generic point X_P along the utopia line a line (line NU) is drawn perpendicularly. The feasible design space is then reduced to the shaded region of Fig. 2b, and the objective μ_2 is minimized yielding the displayed Pareto solution. To generate a set of Pareto solutions, we simply increment the position of X_P along the utopia line. In general, an even distribution of points X_P will result in a corresponding set of evenly distributed Pareto solutions, subject to some scaling issues that are beyond the scope of this cursory presentation of the method. Messac et al.[29] present an important extension of the normal constraint method that addresses scaling issues.

2. *Generic n-Objective Formulation of the Normal Constraint Method*

We begin the presentation of the generic case by defining quantities that are used in the formulation, namely, the utopia plane, the normalized shifting increment, and the point X_P. The utopia plane, which was the utopia line in the preceding biobjective description, is defined by n anchor points (μ^{1*} through μ^{n*}). The utopia plane is defined by $n-1$ vectors given by

$$N_i = \mu^{n*} - \mu^{i*} \qquad i = 1, \ldots, n-1 \tag{7}$$

The normalized increment, through which the line NU is shifted along the utopia plane in the direction of N_i, is given as

$$\delta_i = 1/(m_i - 1), \qquad i = 1, \ldots, n-1 \tag{8}$$

The point X_P, through which the line NU passes perpendicularly, is given by

$$X_{Pj} = \sum_{i=1}^{n} \alpha_{ij} \mu^{i*} \tag{9}$$

where $0 \le \alpha_{ij} \le 1$ and

$$\sum_{i=1}^{n} \alpha_{ij} = 1$$

The normal constraint problem formulation for the n-objective case is given by problem 3.

Problem 3 (P3): Normal constraint method for finding Pareto solutions (n-objective case)

$$\min_x \mu_n(x) \tag{10}$$

Subject to:

$$g_q(x) \le 0, \qquad q = 1, \ldots, r \tag{11}$$

$$h_j(x) = 0, \qquad j = 1, \ldots, v \tag{12}$$

$$N_i(\mu - X_{Pj})^T \le 0, \qquad i = 1, \ldots, n-1 \tag{13}$$

$$x_{il} \le x_i \le x_{iu}, \qquad i = 1, \ldots, n_x \tag{14}$$

where

$$\mu = [\mu_1(x) \cdots \mu_n(x)]^T \qquad (n \ge 2) \tag{15}$$

Problem 3 is then solved for each X_{Pj} [Eq. (9)], resulting in a set of Pareto solutions that collectively represent the Pareto frontier.

The most noteworthy benefits of the normal constraint method are that 1) it generates well-distributed Pareto solutions along the Pareto frontier, 2) it generally converges on fewer non-Pareto solutions than other methods, and 3) it generates well-distributed Pareto solutions even when there is a large difference in design objective scaling.[30] A current limitation of the normal constraint method is that it might miss some peripheral regions of the Pareto frontier because of the conditions immediately following Eq. (9).

While generating the Pareto frontier, an unfortunate feature of this and other methods is that some generated solutions might not actually be globally Pareto optimal. Instead, some solutions might be locally Pareto optimal or non-Pareto optimal. In these circumstances we use a Pareto filter that is designed to eliminate all dominated solutions and retain only those that are globally Pareto optimal. We formally introduce the Pareto filter in Sec. II.C.

C. Development of a Pareto Filter

As just discussed, we develop a Pareto filter to remove spurious design alternatives resulting from Pareto frontier generators. The filter 1) removes all non-Pareto solutions, 2) removes all locally Pareto solutions, and 3) retains all globally Pareto solutions. For the development of the filter, we use the following definitions (for a minimization problem).

1. *Global Pareto Optimality*

A design objective vector μ^* is globally Pareto optimal if another design objective vector μ does not exist in the feasible design space such that $\mu_i \le \mu_i^*$ for all $i \in \{1, 2, \ldots, n\}$ and $\mu_j < \mu_j^*$ for at least one index of j, $j \in \{1, 2, \ldots, n\}$.

2. *Local Pareto Optimality*

A design objective vector μ^* is locally Pareto optimal if another design objective vector μ does not exist in the neighborhood of μ^* such that $\mu_i \le \mu_i^*$ for all $i \in \{1, 2, \ldots, n\}$ and $\mu_j < \mu_j^*$ for at least one index of j, $j \in \{1, 2, \ldots, n\}$ (Ref. 8).

The Pareto filter algorithm compares each design solution with every other generated design solution. When a point is deemed not globally Pareto optimal, it is eliminated. The steps of the filter are presented in the flow diagram of Fig. 3. The four steps of the filter algorithm are described as follows:

1) Initialize the algorithm indices and variables: $i = 0$, $j = 0$, $k = 1$, and $m =$ number of generated solutions.

2) Set $i = i + 1$; $j = 0$.

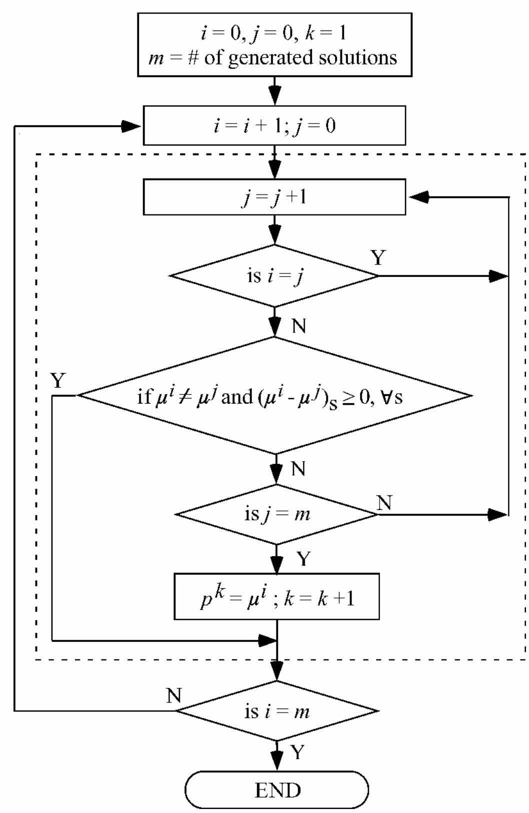


Fig. 3 Pareto filter flow diagram.

3) Eliminate nonglobal Pareto points by the following (see dashed box in Fig. 3): Set $j = j + 1$. If $i = j$, then go to the beginning of step 3, else continue. If $\mu^i \neq \mu^j$ and $(\mu^i - \mu^j)_s \geq 0, \forall s$, then go to step 4 ($\mu_i$ is not a global Pareto point); else if $j = m$, then μ_i is a global Pareto point. Set $p^k = \mu^i$, $k = k + 1$, and go to step 4, else go to the beginning of step 3.

4) If $i \neq m$, go to step 2, else end.

When the algorithm ends, the matrix P, composed of p^k, will have a set of globally Pareto optimal solutions.

III. Concept Selection Using s-Pareto Frontiers

In this section we present a Pareto-based decision-making approach for concept selection in engineering design. Under the approach s-Pareto frontiers are used to assess tradeoffs between design concepts and to perform concept selection. In Sec. III.A a description of s-Pareto optimality is given. In Sec. III.B a mathematical problem statement for obtaining the s-Pareto frontier is presented. Finally, Sec. III.C discusses decision making and concept selection with s-Pareto frontiers.

A. s-Pareto Optimality

Recall the biobjective minimization problem and its associated feasible region and Pareto frontier as shown in Fig. 2a. In general, the solutions on the Pareto frontier represent optimal alternatives to a single design concept. Figure 2a is a typical representation of how Pareto frontiers are used by designers to assess tradeoffs between design alternatives. It is important to observe that under this traditional framework issues concerning the evaluation of a set of design concepts do not typically come to the fore because, generally, only one concept is being evaluated.

Consider now the use of Pareto frontiers earlier in the design process, prior to the selection of a unique concept. Figure 2c represents three design concepts and their respective Pareto frontiers. The shaded areas are the feasible design regions, and the heavy black lines are the Pareto frontiers for each concept. In Fig. 2c we see that the individual Pareto frontiers for concepts A, B, and C are convex. If instead these frontiers were nonconvex, the discussion up to this point would remain unchanged.

For discussion simplicity, we consider a biobjective minimization case with three design concepts. We note, however, that this development fully and directly applies to problems of n objectives and p concepts. In the following development we examine a single optimal solution for each concept by using the AOF given by Eq. (16), where w_1 and w_2 are scalar weights of values 1 and 2, respectively:

$$\min_x J = w_1\mu_1(x) + w_2\mu_2(x) \tag{16}$$

Although the weighted sum formulation of Eq. (16) suffers from serious drawbacks, as discussed in Sec. I, its simplicity facilitates the current discussion. Even in the simplest of practical cases, we do not recommend the use of the weighted-sum method for engineering design optimization, when nonconvex Pareto frontiers are potentially present. Instead, we suggest that one carefully choose another formulation for the AOF,[18,31] such as the weighted square sum, compromise programming, physical programming,[32] or goal programming formulations.

Figure 2d shows the minima μ^{*A}, μ^{*B}, and μ^{*C} of Eq. (16) for each concept A, B, and C, respectively. By taking the minimum AOF value for the set $[\mu^{*A}, \mu^{*B}, \mu^{*C}]$, the s-Pareto solution, or optimum for the set of concepts, emerges for the specified weights. In the case shown in Fig. 2d, μ^{*B} is not only optimum for concept B, but also for the entire set of concepts under evaluation. We call this an s-Pareto solution because it is the Pareto solution for the entire set of concepts, given the specified weights.

Now, let us consider the three concept-specific frontiers of Fig. 2c collectively and eliminate the solutions that are dominated with respect to other solutions in the set of concepts. Upon performing this elimination, the resulting frontier (shown in Fig. 2e) is now Pareto optimal with respect to the set of concepts, thus the name *s-Pareto frontier.* The significance of the s-Pareto frontier is that it makes it possible to use optimization to explore the design space in the early phases of design, as it pertains to more than one concept. For example, it can be seen that for lower values of μ_2 concept B is the optimum. Likewise, for lower values of μ_1 concept A is the optimum. Concept C is never the optimum, regardless of which objective has a higher minimization priority.

An s-Pareto frontier can be used to classify each concept based on its dominance disposition. Concepts can be classified as dominant, partially dominant, or dominated. Concept k is dominant if its Pareto frontier is the s-Pareto frontier. Concept k is dominated if no part of its Pareto frontier is part of the s-Pareto frontier. Concept k is partially dominant if it is neither dominant nor dominated. We note that this use of Pareto optimality is a marked departure from its traditional use of merely choosing a final design alternative.

Exactly what algorithmic implementation approach is used to generate the s-Pareto frontier is not part of the development of the notion of s-Pareto optimality. In particular cases, the weighted-sum approach described earlier might be appropriate. A more robust approach using the normal constraint method is presented later and used in an example.

We make an interesting and important observation at this point. The computations of the design metrics μ_1 and μ_2 are generally different for concepts A, B, and C, but the resulting values represent the same physical quantity. For example, assume that μ_1 is mass in kilograms. The computational formulation used to evaluate mass for concept A might be completely different from that used to evaluate mass for concept B. Likewise, design variables might be completely unique to each concept. This is because the concepts are not mere variations of each other. To make their comparison logical and meaningful, we evaluate these concepts based on the same design metrics. Another important observation is that, although in this case the Pareto frontiers for concepts A and B are individually convex, the associated s-Pareto frontier is concave.

We now comment on the availability of analytical models that describe the concepts' performance, during the conceptual design phase. Importantly, we note that during concept selection, the fidelity of analytical models is low, when they are available at all. However, we recognize that, even when used implicitly, some form of modeling is used to estimate the anticipated performance of considered concepts. In cases where no quantitative model is available, or used, the decision-making process is indeed precarious; and the designer is forced to use less rigorous approaches for concept selection. It is in fact this unfortunate scenario of haphazard decision making at the most critical phase of the design process that this paper seeks to address. Accordingly, we assume that at least rudimentary models of the concepts' performance are available.

B. Generating the s-Pareto Frontier

In this paper the method for generating s-Pareto frontiers assumes that the following information is available: 1) a set of design concepts (to be evaluated), 2) design objectives that are used to evaluate a set of design concepts (called *set objectives*), 3) design objectives that are specific to one design concept (called *concept objectives*), and 4) top-level designer preferences associated with concept objectives.

We now define the difference between set objectives and concept objectives. Set objectives apply to the entire set of concepts under evaluation. Concept objectives apply to one or some (but not all) concepts in the set.

Set objectives are used to guide concept generation and are hereby the basis for comparison during concept selection. As such, set objectives are considered to be known by the designer at the beginning of the concept selection process. This assumption is fully sensible, as these objectives are the basis for the design at a fundamental level, regardless of the concept (e.g., minimize mass, maximize profit).

Concept objectives are often the result of specialized design concepts that need to meet objectives which are particular to that given concept. For example, a particular design concept might be called the "low-cost concept" because its design specifically targets the minimization of cost. Similarly, another design concept might be particularly vulnerable to safety issues. In that particular case, we might be interested in maximizing a safety objective for this concept only, or perhaps include this concept objective as an inequality behavioral constraint in the problem formulation. It is assumed that concept objectives are known by the designer as concept selection begins. Alternatively, these objectives can be included at an intermediate stage of the concept selection process, as knowledge about each concept evolves. Because concept objectives cannot be ignored, we include them in the selection process by presenting an optimization problem formulation that captures the effect of such design objectives.

To generate the s-Pareto frontier, we use the normal constraint method because it is particularly well suited for design space exploration as it pertains to multiple design concepts. We start by developing an extension to the normal constraint method, so as to make it applicable to multiple design concepts. Concept objectives are then optimized to identify achievable performance (problem 4) and take the form of constraints in the s-Pareto formulation (problem 5). Figure 2f shows the basic application of the normal constraint method for generating s-Pareto solutions.

We find anchor points by solving P2 for each design concept. The anchor points are then used to identify the endpoints of the s-Pareto frontier, which are called *s-anchor points* and are denoted as μ^{si*} for $i = 1, \ldots, n$. The s-utopia plane is defined by $n-1$ vectors given by

$$N_i = \mu^{sn*} - \mu^{si*} \tag{17}$$

Because the N_i direction is defined by the s-anchor points, the utopia plane is the same for all design concepts under evaluation. As a result, the design space as a whole will be well represented by s-Pareto solutions (i.e., an even distribution of s-Pareto solutions is generated).

In the following problem statement (problem 4) concept objectives are optimized for each concept k. This is done because concept objectives are not otherwise a direct part of the evaluation process. Instead, the influence of concept objectives is captured in the evaluation by an added constraint. This added constraint is obtained by solving problem 4 and can become part of problem 5 [Eq. (28)]. From problem 4 we identify achievable performance. With this information, and an allowable relaxation variable [Eq. (28)], we can appropriately formulate a problem to generate the s-Pareto frontier (problem 5).

Problem 4 (P4): Optimization of concept objectives for concept k

$$J^{*ck} = \min_{x^k} \; J^{ck}(\mu^{ck}) \tag{18}$$

Subject to:

$$g_q^{ck}(x^k) \le 0, \qquad q = 1, \ldots, r \tag{19}$$

$$h_j^{ck}(x^k) = 0, \qquad j = 1, \ldots, v \tag{20}$$

$$x_{il}^k \le x_i^k \le x_{iu}^k, \qquad i = 1, \ldots, n_{x^k} \tag{21}$$

where

$$\mu^{ck} = \left[\mu_1^{ck}(x^k) \; \cdots \; \mu_n^{ck}(x^k)\right]^T \qquad (n \ge 0) \tag{22}$$

$$x^k = \left[x_1^k \; \cdots \; x_{n_x^k}^k\right]^T \tag{23}$$

and k, $1 \le k \le p$, denotes the kth concept, μ^{ck} is a vector of concept objectives for concept k, and x^k is the design variable vector for concept k. P4 calls for defining top-level designer preference for some design objectives of concept k. As the designer preferences are specified for P4, a unique optimal solution is obtained. As discussed earlier, the solution to P4 can be used as a constraint in problem 5.

We now generate Pareto solutions for each design concept by solving problem 5. We do this in the manner depicted in Fig. 2f. The objective of problem 5 is to find the concept with the smallest minimized AOF value. To let concept k represent an infinite number of alternatives, a relaxation/slack variable is included in Eq. (28).

Problem 5 (P5): Generation of s-Pareto solutions

$$\min_k \left\{ \min_{x^k} \mu_n^k \right\} \tag{24}$$

Subject to:

$$g_q^k(x^k) \le 0, \qquad q = 1, \ldots, r \tag{25}$$

$$h_j^k(x^k) = 0, \qquad j = 1, \ldots, v \tag{26}$$

$$N_i\left(\mu^k - X_{Pj}\right)^T \le 0, \qquad i = 1, \ldots, n-1 \tag{27}$$

$$J^{ck}(\mu^{ck}) \le J^{*ck} + T^k \tag{28}$$

$$x_{il}^k \le x_i^k \le x_{iu}^k, \qquad i = 1, \ldots, n_x \tag{29}$$

where

$$\mu^k = \left[\mu_1^k(x^k) \cdots \mu_n^k(x^k)\right]^T \qquad (n \ge 2) \tag{30}$$

$$x^k = \left[x_1^k \cdots x_{n_x^k}^k\right]^T \tag{31}$$

and k, $k \le p$, denotes the kth concept, μ^k is a vector containing design metrics for concept k, the terms in Eq. (27) are defined by Eqs. (9) and (17), x^k denotes the design variable vector for concept k, and J^{*ck} is defined as the optimal AOF value from P4. This slack variable is designed to capture different degrees of stringency placed on the minimization of concept objectives. Figure 4 shows that by relaxing the minimization of μ_3 a Pareto frontier for μ_1 and μ_2 can be obtained. Problem 5 is repeated for each X_{Pj}, resulting in a set

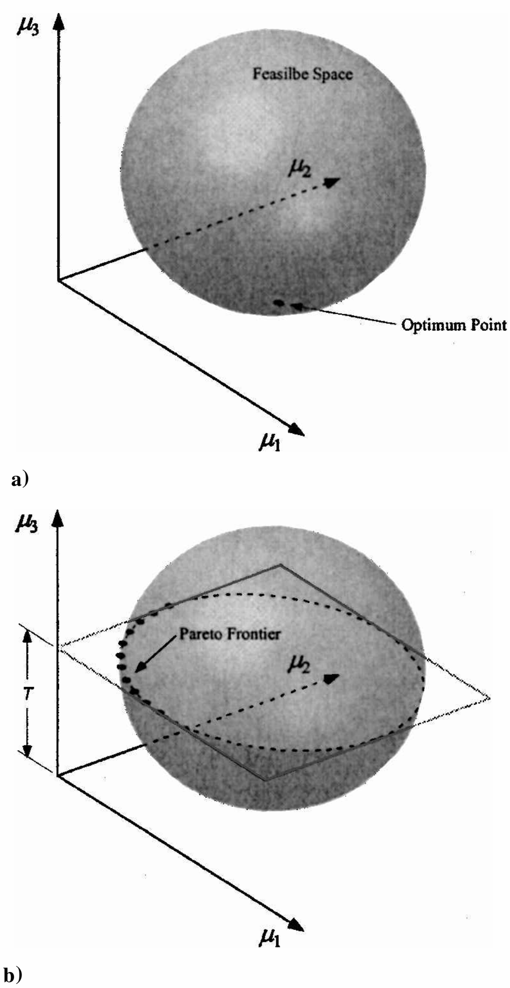


Fig. 4 Slack variable used to relax the stringency of minimizing μ_3.

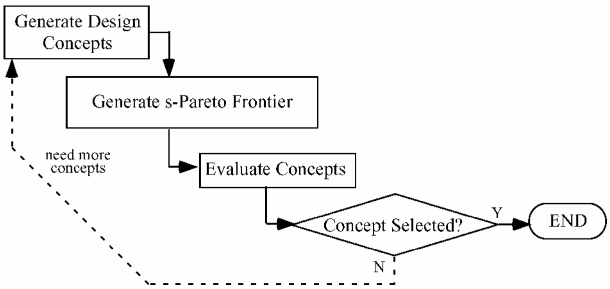


Fig. 5 Pareto-based concept selection flowchart.

of s-Pareto solutions. We then employ the use of the Pareto filter to remove all solutions that are not s-Pareto optimal. Consequently, the remaining set discretely represents the s-Pareto frontier.

C. Decision Making with s-Pareto Frontiers

In this section we examine design decisions that can be made using the s-Pareto frontier. The basic approach to using s-Pareto frontiers to perform concept selection can be represented by the flowchart in Fig. 5. The process starts (top left box) with candidate design concepts that result from concept generation activities. A Pareto frontier is generated for each design concept, and the s-Pareto optimal solutions are identified resulting in the s-Pareto frontier. Various concepts are then compared (evaluated) based on their position within the design space, or their dominance disposition.

Consider the three concepts that are shown in Fig. 2g. The original feasible regions for each concept are represented by closed loops. Also shown in Fig. 2g are six region indicators (1–6) that describe six different regions of the design space. For example, the bounds of the region indicated by marker number 2 are delineated by the dashed lines. The design space to the southwest of the dashed lines represents the region of interest. Under the proposed method we will evaluate the design concepts only within the given region (southwest of the region indicator). We call these regions *regions of interest* (RI_i).

As indicated in the flowchart, after generating the s-Pareto frontier the candidate design concepts are evaluated. This evaluation is based on a specified region of interest, that is, concepts are evaluated based on their position within a region of interest. In the case of RI_2, concept A is the only concept within the region and is therefore the preferred concept. Even though it is the preferred concept, the designer can reasonably choose to 1) end the concept selection process by selecting concept A, 2) reevaluate the concepts based on a different region of interest, or 3) generate additional design concepts.

We now briefly discuss RI_1, RI_3–RI_6, which are indicated by markers 1, 3–6. Within RI_1 there are no design concepts so the designer can 1) reevaluate at a different desired region or 2) return to concept generation. RI_4 indicates that concept B is preferred. In this case the designer faces the same general options as faced with RI_2. Within RI_3, RI_5–RI_6, the designer needs to choose between two concepts. Because both concepts are partially dominant, the final selection depends fully on the designer's preferences. If the designer is indifferent to the two concepts, then either or both can be pursued. It is also possible for the designer to evaluate the amount of space (length of the Pareto frontier, for the biobjective case) each concept occupies within the desired region. For example, consider RI_5 and the amount of space occupied by concept A as compared to that of concept B. Concept A clearly occupies a larger amount of space in the region of interest. A designer can conclude that concept A offers more flexibility within the region of interest than concept B does and can therefore choose to select concept A.

Concept C should not be chosen as the preferred concept. This is simply because concept C is dominated by both concepts A and B. Even though RI_6 contains the Pareto frontier for concept C, the frontier for concept C is not part of the s-Pareto frontier and therefore should not be chosen.

Finally, we make the important observation that, unlike decision-matrix-based methods where concept selection is based on a single performance number for objectives, the s-Pareto approach accounts for the objectives behaviors over ranges.

IV. Truss Design Example

To illustrate the usefulness of s-Pareto frontier-based concept selection, we consider an example often used in the optimization literature: optimization of a three-bar truss. Figure 6a is a diagram of the simple truss structure. This truss example was originally introduced by Koski[9] and later revisited by others.[10,11,28] The truss design problem seeks to obtain an optimal value of b, and cross-sectional areas of each member, that minimize the nodal displacement at node P and the total volume of the structure, while supporting horizontal and vertical loads.

From a design-process perspective this is a detailed design, or design refinement, problem. We say this because the general design configuration (three bar truss, with a vertical member in the center, etc.) has already been chosen. The approach taken by others who have used this three-bar truss example has been to optimize the truss in Fig. 6a without explicit consideration as to why it, of all configurations, is best.

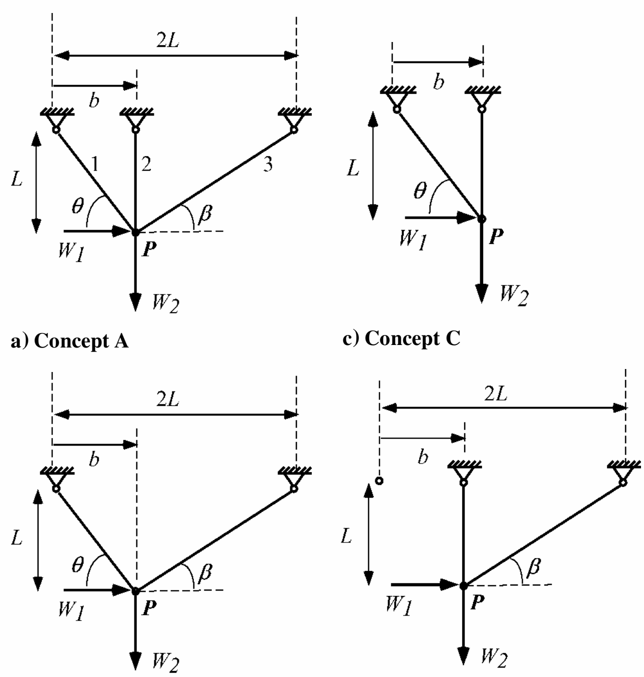
We consider instead this same example from a functional design perspective; a structure to support horizontal and vertical loads of 100 kips (0.445 MN) and 1000 kips (4.45 MN), respectively, must be designed. Its height is to be no more than L and length no more than $2L$. The horizontal location of node P, b, must be between $0.5L$ and $1.5L$. Given this scenario, a design team might generate a number of concepts such as those shown in Fig. 6.

Given concepts A–D, we will now show how s-Pareto frontiers can be used to perform concept selection. In Sec. IV.A we provide the truss optimization problem statement. In Sec. IV.B the s-Pareto frontier for this set of concepts is generated, and finally, concept selection is performed in Sec. IV.C.

A. Truss Problem Formulation

The optimization problem statement for the truss design example is given as follows:

$$\min_{a,b}\begin{Bmatrix}\mu_1(a,b)\\ \mu_2(a,b)\end{Bmatrix} = \min_{x}\begin{Bmatrix}\text{Displacement squared}\\ \text{Volume}\end{Bmatrix} \tag{32}$$



a) Concept A
c) Concept C
b) Concept B
d) Concept D

Fig. 6 Multiple concepts for truss design example.

Subject to:

$$\tan\theta = L/b \tag{33}$$

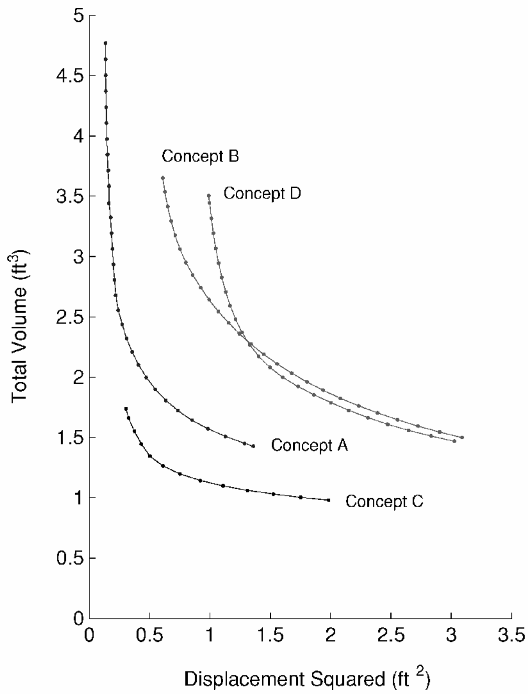
$$\tan\beta = L/(2L - b) \tag{34}$$

$$\sigma_i \leq \sigma_{\max}, \qquad i = 1, 2, 3 \tag{35}$$

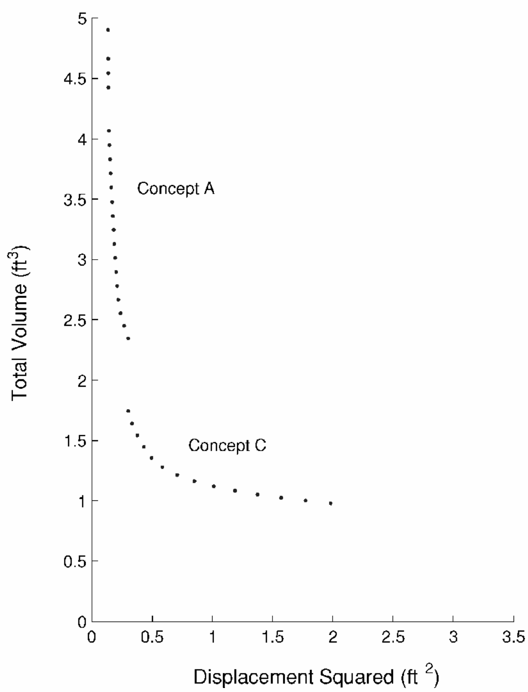
$$0.8\,\text{in.}^2(5.16\,\text{cm}^2) \leq a_i \leq 3\,\text{in.}^2(19.35\,\text{cm}^2), \qquad i = 1, 2, 3 \tag{36}$$

$$L/2 \leq b \leq 3L/2 \tag{37}$$

The equality constraints [Eqs. (33) and (34)] are included to ensure that the structure remains connected at node P. The stress in each bar must be lower than the maximum allowable stress, as indicated by Eq. (35), where the bar on the left (Fig. 6a) is bar 1, the vertical bar is bar 2, and the bar on the right is bar 3. The cross-sectional area of each bar a_i is limited as described by Eq. (36), and the horizontal location of node P, b, is also constrained by Eq. (37). The fixed parameters for this problem are defined as follows: Young's modulus E is 29×10^3 ksi (200 GPa); truss dimension L is 60 ft (18.3 m); the maximum allowable stress $\sigma_{\max}$ is 550 ksi (3.8 GPa); and the loads W_1 and W_2 are 100 kips (0.445 MN) and 1000 kips (4.45 MN), respectively.



a)



b)

Fig. 7 Pareto frontiers for each truss concept and the s-Pareto frontier for the set of concepts.

B. Generating the s-Pareto Frontier for the Set of Truss Concepts

We now use the information given in Eqs. (32–37) to solve P5 and generate an s-Pareto frontier for the set of truss concepts. Let us first discuss the individual Pareto frontiers for each truss concept, as shown in Fig. 7a.

After identifying the s-Pareto frontier, it can be seen that none of the Pareto solutions from concepts B and D are part of the s-Pareto frontier. It can also be seen that some solutions from concept A are part of the s-Pareto frontier and some are not. Likewise, all solutions from concept C are part of the s-Pareto frontier. All of the s-Pareto solutions, from the set of truss concepts, can then be joined together to form the s-Pareto frontier as shown in Fig. 7b. This frontier is generated using the formulation given in problem P5 and through applying the Pareto filter.

C. Selecting Truss Concepts

From the s-Pareto frontier we can conclude that concepts B and D are classified as dominated because none of their individual Pareto solutions are part of the s-Pareto frontier. Likewise, concept A is classified as partially dominant as only a portion of its solutions are part of the s-Pareto frontier. Concept C is also classified partially dominant because only a portion of the s-Pareto frontier is from concept C. From this evaluation it is clear that concepts A and C merit further consideration, whereas concepts B and D do not. Further examination through exploring specific regions of interest can lead to a better understanding of the flexibility associated with each design. This point is discussed in Sec. IV.C.

The power and usefulness of this s-Pareto frontier is that it can be used to assess tradeoffs between design concepts during a stage in the design process when decisions have a large impact on the success of the design (Fig. 1). For example, if the square of the nodal displacement needs to be less than 0.25 ft^2 then concept A is the concept of choice. If instead, the total volume needs to be below, say 1.5 ft^3, then concept C is the concept of choice. We have shown each concept's Pareto frontier in Fig. 7a for the sake of discussion. However, when one solves P5, Fig. 7b is obtained directly.

V. Conclusions

In this paper we have presented a method for concept selection using the proposed notion of s-Pareto optimality. In particular, the method focuses on the generation and use of s-Pareto frontiers to classify design concepts as dominant, partially dominant, or dominated. This paper shows that such classifications facilitate the selection of concepts that merit further consideration and ultimately the selection of a single optimal concept. We have examined a truss design problem and have shown how Pareto optimality can be used in the context of multiple design concepts and that s-Pareto frontiers make possible a computational optimization approach to concept selection in the conceptual phase of design.

Acknowledgments

This research was partially supported by National Science Foundation Grant DMI-0196243 for Achille Messac. We acknowledge the members of the Multidisciplinary Design and Optimization Laboratory at Rensselaer Polytechnic Institute for the collaborative effort in the development the Pareto filter.

References

[1]Ishii, K., “Life-Cycle Engineering Design,” *Journal of Mechanical Design*, Vol. 117, June 1995, pp. 42–47.

[2]Homan, B. S., and Thornton, A. C., “Precision Machine Design Assistant: A Constraint-Based Tool for the Design and Evaluation of Precision Machine Tool Concepts,” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, Vol. 12, No. 5, 1998, pp. 419–429.

[3]Wang, J., “Ranking Engineering Design Concepts Using a Fuzzy Outranking Preference Model,” *Fuzzy Sets and Systems*, Vol. 119, 2001, pp. 161–170.

[4]Li, H., and Azarm, S., “Product Design Selection Under Uncertainty and with Competitive Advantage,” *Journal of Mechanical Design*, Vol. 122, 2000, pp. 411–418.

[5]Crossley, W. A., Martin, E. T., and Fanjoy, D. W., “Conceptual Aircraft Design Environment: Case Study Evaluation of Computing Architecture Technologies,” AIAA Paper 2001-5247, Oct. 2001.

[6]Simpson, T. W., Rosen, D., Allen, J. K., and Mistree, F., “Metrics for Assessing Design Freedom and Information Certainty in the Early Stages of Design,” *Journal of Mechanical Design*, Vol. 120, 1998, pp. 628–635.

[7]Steuer, R. E., *Multiple Criteria Optimization, Theory Computations and Applications*, Wiley, New York, 1986, Chap. 3.

[8]Miettinen, K. M., *Nonlinear Multiobjective Optimization*, International Series in Operations Research and Management Science, Kluwer Academic, Norwell, MA, 1999, pp. 29–43.

[9]Koski, J., “Multicriteria Truss Optimization,” *Multicriteria Optimization in Engineering and in the Sciences*, edited by W. Stadler, Plenum, New York, 1988, pp. 263–306.

[10]Das, I., “Nonlinear Multicriteria Optimization and Robust Optimality: A Truss Optimization Problem,” Ph.D. Dissertation, Rice Univ., Houston, TX, May 1997, Chap. 9.

[11]Belegundu, A., and Chandrupatla, T., *Optimization Concepts and Applications in Engineering*, Prentice–Hall, Upper Saddle River, NJ 1999, pp. 373–382.

[12]Grandhi, R. V., and Bharatram, G., “Multiobjective Optimization of Large-Scale Structures,” *AIAA Journal*, Vol. 31, No. 7, 1993, pp. 1329–1337.

[13]Edgeworth, F. Y., *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*, Kegan Paul, London, 1881.

[14]Pareto, V., *Cour d’Economie Politique*, Librarie Droz, Geneva, 1964.

[15]Koopmans, T. C., “Analysis and Production as an Efficient Combination of Activities,” *Activity Analysis of Production and Allocation*, edited by T. C. Koopmans, Yale Univ. Press, New Haven, CT, 1951.

[16]Pahl, G., and Beitz, W., *Engineering Design: A Systematic Approach*, 2nd ed., Springer-Verlag, London, 1996, pp. 99–117.

[17]Ulrich, K. T., and Eppinger, S. D., *Product Design and Development*, 2nd ed., McGraw–Hill, New York, 2000, pp. 138–159.

[18]Messac, A., and Ismail-Yahaya, A., “Required Relationship Between Objective Function and Pareto Frontier Orders: Practical Implications,” *AIAA Journal*, Vol. 39, No. 11, 2001, pp. 2168–2174.

[19]Das, I., and Dennis, J., “A Closer Look at Drawbacks of Minimizing Weighted Sums of Objective for Pareto Set Generation in Multicriteria Optimization Problems,” *Structural Optimization*, Vol. 14, 1997, pp. 63–69.

[20]Das, I., and Dennis, J., “Normal-Boundary Intersection: A New Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems,” *SIAM Journal on Optimization*, Vol. 8, No. 3, 1998, pp. 631–657.

[21]Koski, J., “Defectiveness of Weighting Methods in Multicriterion Optimization of Structures,” *Communications in Applied Numerical Methods*, Vol. 1, 1985, pp. 333–337.

[22]Pugh, S., *Creating Innovative Products Using Total Design: The Living Legacy of Stuart Pugh*, 1st ed., Addison Wesley Longman, Reading, MA, 1996, pp. 167–182.

[23]Patel, M., Lewis, K., Messac, A., and Maria, A., “Subsystem Design Through Subsystem Selection Using Physical Programming, *AIAA Journal* (to be published).

[24]Perez, R. E., Chung, J., and Behdinan, K., “Aircraft Conceptual Design Using Genetic Algorithms,” AIAA Paper 2000-4938, Sept. 2000.

[25]Yang, R. J., Chuang, C.-H., and Che, X., “Recent Applications of Topology Optimization,” AIAA Paper 98-4849, Sept. 1998.

[26]Chin, K.-S., and Wong, T., “Knowledge-Based Evaluation for the Conceptual Design Development of Injection Molding Parts,” *Engineering Applications in Artificial Intelligence*, Vol. 9, No. 4, 1996, pp. 359–376.

[27]Ismail-Yahaya, A., and Messac, A., “Effective Generation of the Pareto Frontier: Normalized Normal Constraint Method,” AIAA Paper 2002–1232, April 2002.

[28]Messac, A., and Mattson, C. A., “Generating Well-Distributed Sets of Pareto Points for Engineering Design Using Physical Programming,” *Optimization and Engineering*, Vol. 3, Dec. 2002, pp. 431–450.

[29]Messac, A., Ismail-Yahaya, A., and Mattson, C. A., “The Normalized Normal Constraint Method for Generating the Pareto Frontier,” *Structural and Multidisciplinary Optimization* (to be published).

[30]Ismail-Yahaya, A., and Messac, A., “Effective Generation of the Pareto Frontier Using the Normal Constraint Method, AIAA Paper 2002-0178, Jan. 2002.

[31]Messac, A., From the Dubious Construction of Objective Functions to the Application of Physical Programming, *AIAA Journal*, Vol. 38, No. 1, 2000, pp. 155–163.

[32]Messac, A., “Physical Programming: Effective Optimization for Computational Design,” *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 149–158.

E. Livne
Associate Editor