Smart Pareto Filter: Obtaining a Minimal Representation of Multiobjective Design Space

Christopher A. Mattson

Anoop A. Mullur

Achille Messac

Corresponding Author

Achille Messac, Ph.D.
Distinguished Professor and Department Chair
Mechanical and Aerospace Engineering
Syracuse University, 263 Link Hall
Syracuse, New York 13244, USA

Email: messac@syr.edu
Tel: (315) 443-2341
Fax: (315) 443-3099
https://messac.expressions.syr.edu/

Bibliographical Information

Smart Pareto Filter: Obtaining a Minimal Representation of Multiobjective Design Space

Christopher A. Mattson, Anoop A. Mullur and Achille Messac

Multidisciplinary Design and Optimization Laboratory, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA

(Received 11 February 2004; In final form 3 June 2004)

Multiobjective optimization is a powerful tool for resolving conflicting objectives in engineering design and numerous other fields. One general approach to solving multiobjective optimization problems involves generating a set of Pareto optimal solutions, followed by selecting the most attractive solution from this set as the final design. The success of this approach critically depends on the designer’s ability to obtain, manage, and interpret the Pareto set—importantly, the size and distribution of the Pareto set. The potentially significant difficulties associated with comparing a significantly large number of Pareto designs can be circumvented when the Pareto set: (i) is adequately small, (ii) represents the complete Pareto frontier, (iii) emphasizes the regions of the Pareto frontier that entail significant tradeoff, and (iv) de-emphasizes the regions corresponding to little tradeoff. We call a Pareto set that possesses these four important and desirable properties a smart Pareto set. Specifically, a smart Pareto set is one that is small and effectively represents the tradeoff properties of the complete Pareto frontier. This article presents a general method to obtain smart Pareto sets for problems of n objectives, given previously generated sets of Pareto solutions. Under the proposed method, the designer uses a smart Pareto filter to control the size of the Pareto set and the degree of tradeoff representation among objectives. Importantly, the smart Pareto filter yields a Pareto set comprising a minimal number of solutions needed to adequately characterize the problem’s tradeoff properties. In this article, the smart Pareto filter is analytically developed, and mathematical and physical examples are presented to illustrate the filter’s effectiveness.

Keywords: Multiobjective optimization; Pareto filter; Pareto optimality; Smart Pareto filter

1 Introduction and Literature Survey

Engineering design generally involves assessing and managing tradeoffs between conflicting design objectives. Multiobjective optimization is a powerful tool that is commonly used to resolve such conflicts and arrive at a final design. Approaches for solving multiobjective optimization problems can be divided into two broad categories. In the first, termed Integrated Generating–Choosing (IGC) [1], the designer forms an aggregate objective function (AOF) that is intended to encapsulate all design objectives and the designer’s preference with regard to handling the tradeoffs between them. The optimization of this AOF, subject to relevant constraints, is expected to yield an optimal solution when this approach succeeds. The main
difficulty of this approach is in forming the correct AOF. For realistic problems, forming the AOF is a significant challenge. The physical programming method is one that explicitly addresses this challenge [2].

Either on the basis of personal preference or the nature of the problem, the designer may follow a second approach. Under this alternate approach (termed the Generate First–Choose Later (GF–CL) [1, 3]), a set of candidate designs is generated and the most desirable among them is chosen as a final design. The success of this approach critically depends on the designer’s ability to obtain, manage, and interpret the set of candidate designs. This article introduces a new methodology that explicitly addresses this critical challenge.

Using either the first or second approach, it is generally desirable for the final solution to be Pareto optimal. Pareto optimal solutions are those for which any improvement in one objective will result in the worsening of at least one other objective. That is, a tradeoff will take place. The concept of Pareto optimality is central to multiobjective optimization [4–7]. The complete set of such solutions is referred to as the Pareto surface, or Pareto hyper-surface – for problems of $n$ objectives. We use the term Pareto set to refer to a discrete representation of the Pareto hyper-surface. Pareto surfaces are also referred to as Pareto frontiers. In order to benefit more from multiobjective optimization using the GF–CL approach, the designer must judiciously obtain the Pareto set. The goodness of the Pareto set is the central focus of this study. Specifically, this article develops a methodology that identifies a minimal number of Pareto solutions needed to adequately represent the multiobjective design space; thus providing the designer with a small number of disparate design alternatives from which to choose the final design.

Before discussing the new methodology, we review the literature in the areas of Pareto-set generation and Pareto-set filtering. This review and corresponding observations are provided in the following sections.

1.1 Pareto-set Generation

The design literature shows a natural evolution of strategies used to generate Pareto sets. The first phase in the evolution involves a general push to obtain any set of Pareto optimal solutions, from which the final design is selected. The second phase illustrates a change in the thinking, as a push towards obtaining an even distribution of points on the Pareto frontier emerges. Under the second phase, the even distribution of Pareto points is viewed as superior because it does not over-represent or under-represent any one part of the Pareto frontier, thus allowing the designer to explore the complete range of the objectives values. Under the new approach presented in this article, the thinking evolves yet again as a third phase in the evolution is entered. Specifically, we propose the notion that it is desirable to obtain a small (minimal) set of Pareto solutions that adequately represents the tradeoff properties of the complete Pareto frontier – be it evenly distributed or not. Each of these three phases is now discussed.

1.1.1 Phase 1: Obtaining Any Pareto Set

Various approaches have been developed for obtaining Pareto sets [1, 8–11]. One of the most common approaches is to repeatedly solve an optimization problem, each time using a systematically varied set of weights in an AOF. Using the weighted-sum AOF is one such possibility where the pre-multiplying factors of each objective are varied, yielding different Pareto solutions. Despite its common use, the weighted-sum method possesses serious limitations and drawbacks, which are well-documented in the literature [12–15]. Generating a set of Pareto solutions using any haphazardly chosen method may entail serious consequences. Specifically, the resulting set may over-represent some regions of the Pareto frontier, while under-representing other regions, and may do so in a way that is detrimental to the design process.
For example, under the weighted sum or the compromise programming approaches, a uniform change in the AOF’s weighting factors does not typically result in a set of evenly spaced Pareto solutions. Rather, it typically results in a cluster of solutions in some regions of the Pareto frontier. A large density of Pareto points in any one region of the Pareto frontier may unduly increase the likelihood of choosing a final solution in that region. Conversely, a small density of points decreases that likelihood. Missing solutions in non-convex regions, for example, is a well-known limitation of the weighted-sum method.

Genetic algorithm approaches [16] offer notable possibilities for obtaining the Pareto set. The genetic algorithm differs from the methods discussed above in that it is an evolutionary search technique that requires no gradient information and can obtain multiple optima in parallel. Genetic algorithms are particularly well suited for cases involving discontinuous or multimodal objective functions. Goldberg [17] developed a Pareto-genetic algorithm that uses a Pareto ranking procedure to move a population toward the Pareto frontier. One drawback of genetic algorithms (and all evolutionary algorithms), however, is the high computational cost associated with their use.

1.1.2 Phase 2: Obtaining an Even Distribution of Pareto Solutions

There has been a recent thrust to develop methods for generating a Pareto set that comprises an even distribution of Pareto solutions [8, 10, 11]. A set of points is said to be evenly distributed over a region if no part of that region is over-represented or under-represented in that set of points, compared with other parts. Such a distribution circumvents the difficulties resulting from missed regions of the Pareto frontier, by ensuring that the entire Pareto surface is well represented. That is, no single part of the Pareto surface is under-represented or over-represented – regardless of its degree of desirability. Messac and Mattson [11] discuss the notion of even distribution in detail, and provide a development for measuring distribution evenness.

Three methods have distinguished themselves in their abilities to generate an even distribution – the normal boundary intersection [8], physical programming [1], and the normal constraint [10, 11] methods. A recent publication by Messac et al. [10] contrasts the strengths and weaknesses of each of these three methods as well as those of other popular approaches such as the weighted sum, compromise programming methods, and ε-constraint methods. In terms of its ability to generate an even distribution of Pareto points over the complete Pareto frontier, the normal constraint method is shown to perform favorably [10, 11].

Another notable approach for generating a set of relatively evenly distributed Pareto solutions is developed by Deb et al. [18, 19]. They present a genetic algorithm based multiobjective optimization strategy in which they define a so-called crowding distance parameter. This parameter helps in maintaining diversity in the Pareto set by avoiding clusters of points and isolated points in the design objective space.

1.1.3 Phase 3: Obtaining a Smart Distribution of Pareto Solutions

There are important reasons to consider an additional phase in the evolution of Pareto set generators; a phase where the strategy evolves from the desire to generate an even distribution of Pareto points to that of generating a smart distribution of points.

Assuming that it is possible to generate a set of evenly distributed Pareto solutions, there may still be a need to have an adequate concentration of points in high tradeoff regions of the Pareto frontier in order to make appropriate decisions in these potentially desirable regions. As a result, an even distribution, or concentration, of points over the Pareto frontier
has its own drawbacks in some practical cases. Namely, there will be a higher-than-needed concentration of points in certain less desirable regions. Indeed, some regions are of greater interest than others. In this article, we make the reasonable assumption that regions where there are significant tradeoffs among the objectives are of greater interest than those where the degree of tradeoff is less significant. Stated differently, when the tradeoff is insignificant, we assume that a designer is willing to give up an insignificant amount in one objective to gain significantly in another. It is when the tradeoff is significant that the decision making is difficult and requires special care. In summary, we conclude that in some cases, the best situation is not necessarily an even distribution of Pareto solutions, since it may result in an unduly high number of solutions for consideration – with undue concentration in unwanted regions.

From the above discussion, we observe that in some practical cases, a more useful distribution of Pareto points is one that entails a judiciously variable density of points. We propose that the density should be higher where there is significant tradeoff, and lower where tradeoff is insignificant. We refer to such a distribution as a smart distribution of Pareto points. This article develops an approach that results in smart distributions of Pareto points. The new method allows the designer to express what he or she considers to be a significant or insignificant tradeoff between two specific designs. The method accordingly results in a reduced set of designs – all of which exhibit significant tradeoff. The resulting set is smaller, thus alleviating the designer’s need to compare an unduly large number of designs in order to choose a final design. Under the proposed approach, the smart Pareto set is obtained through filtering an initially large set of Pareto solutions. As such, we briefly review the Pareto-filtering approaches as they are discussed in the literature.

1.2 Pareto-set Filtering

Two distinct classes of Pareto filters are discussed in the literature. The first involves filtering to eliminate all non-Pareto and locally Pareto solutions from a given set of design points. The result is a set of globally Pareto optimal solutions. The second class involves filtering a global Pareto set to remove Pareto solutions that are deemed undesirable or less useful. Both classes are discussed briefly in Sections 1.2.1 and 1.2.2.

1.2.1 Eliminate Non-Pareto and Locally Pareto Solutions

Unfortunately, nearly all approaches for obtaining the Pareto set may result in the generation of non-Pareto or locally Pareto solutions. Numerically filtering a set of points to remove these unwanted solutions has been addressed in the literature. Montusiewicz and Osyczka [20] developed a four-stage optimization process for machine-tool design. In the initial stages, subsystem objectives and overall system objectives are optimized, and in the final stage a Pareto filter is used to remove all non-globally Pareto solutions. Abraham et al. [21] use a series of design filters (including a Pareto filter) to arrive at optimal system designs. They use a sequence of filters in conjunction with an evolutionary algorithm to perform discrete optimization of embedded computer systems.

Cheng and Li [22] introduce a Pareto filter and use it within a genetic algorithm to prevent the loss of Pareto solutions during the evolutionary process. Messac et al. [10] develop a Pareto filter to overcome deficiencies in Pareto set generation methods. Mattson and Messac [9] also use a Pareto filter to reduce the Pareto frontiers from various disparate design concepts into a single Pareto frontier termed the s-Pareto frontier.
1.2.2 Reduce the Set of Pareto Optimal Solutions

Das [23] suggests that some Pareto points should be viewed as superior to others. He approaches Pareto-set reduction by developing a measure of efficiency for each Pareto point. The most efficient solution, the utopia point, is said to be efficient of order 1. Less efficient solutions are said to be efficient of order greater than 1. Das reduces the Pareto set based on the order of efficiency for each point. In another publication, Das [24] provides a problem formulation for identifying the so-called ‘knee’ of the Pareto frontier. The knee is characterized as the region of the curve that protrudes the farthest away from the line connecting the endpoints of the Pareto frontier (for the bi-objective case). The purpose for identifying the knee is to direct the designer to a region in ‘the middle’ of the Pareto frontier. Das suggests that it is from the middle of the frontier that a designer usually chooses a solution. Using the formulation presented, the knee is only characterized for convex Pareto frontiers.

Di Barba [25] suggests that identifying only a few Pareto solutions is often sufficient to represent the Pareto frontier. Di Barba uses an evolutionary optimization method, with a large number of individuals in the initial population, to obtain the Pareto frontier. The author states that Pareto solutions that are weakly Pareto optimal may be removed by filtering. No detailed filtering approach was presented.

As we conclude the introduction of this article, it is helpful to comment on the meaning of closeness of points in the objective (design metric) space. When properly formulated, the goodness or desirability of a design is almost fully represented by its location in the objective space. For example, when two points are near each other in the objective space, that closeness is supposed to indicate the designer’s indifference to their relative goodness. That is, they are of nearly identical desirability to the designer, regardless of their otherwise different features, or potentially different relative locations in design variable space. If such is not the case, then the objective space becomes less useful or meaningful, and the Pareto frontier also loses its value. These observations explain why we do not focus on the design variable space. If a certain geometric feature is desirable (say, a smaller width), then that feature should simply become part of the objective space. In that case, two designs of differing widths will simply not be neighbors in the objective space.

In this article, we develop a framework that uses a series of filtering strategies to obtain a smart representation of the Pareto frontier. Because the smart Pareto set is generally small in size, the proposed approach circumvents the potentially formidable difficulties associated with a significantly large number of design alternatives that must be examined to choose the final design. The remainder of the article is presented in three parts: Section 2 presents the analytical development of the smart Pareto filter, Section 3 gives examples that illustrate the usefulness of the approach, and finally, Section 4 presents concluding remarks.

2 SMART PARETO FILTER: AN ANALYTICAL DEVELOPMENT

In this section, an approach for obtaining a smart representation of the Pareto frontier is developed analytically. For simplicity, this approach is referred to as the smart Pareto methodology. The flowchart presented in Figure 1 describes the basic approach, and is also a general outline for the presentation given in this section. Figure 1 starts with a multiobjective design problem, which is shown as a dashed box on the left side of the figure; this problem defines the inputs (objectives, constraints, and variables) required to optimize the design under a GF–CL strategy. The output of this multiobjective optimization is a set of candidate designs or solutions. As a note, computational optimization does not necessarily need to be used for
this phase in the approach. For example, a designer may generate various optimal design concepts using traditional concept generation techniques [26].

The next major section in the flowchart indicates that the set of candidate optimal designs is filtered to remove all candidates that are not globally Pareto optimal. The set of globally Pareto solutions is then filtered using a so-called smart Pareto filter; shown as the last major step in the flowchart. The output of this step is a smart Pareto set comprising a minimal number of Pareto designs, where each design is significantly different from the other designs. Together, the minimal set of solutions adequately characterizes the design tradeoff possibilities.

In the following sections, each of the three main parts of the smart Pareto methodology is discussed in detail. Specifically, multiobjective optimization (definitions and approaches) is the topic of Sections 2.1 and 2.2; a global Pareto filter is presented in Section 2.3; and in Section 2.4, the smart Pareto filter is introduced.

2.1 Multiobjective Optimization

This section defines the generic multiobjective optimization problem statement and also defines important terms that will be used throughout the developments of this paper. The generic multiobjective optimization problem can be stated as Problem 1.

**Problem 1: generic multiobjective optimization**

\[
\min_{x} \{\mu_1(x) \mu_2(x) \cdots \mu_n(x)\}^T \quad (n \geq 2),
\]

subject to

\[
g(x) \leq 0,
\]
\[
h(x) = 0,
\]
\[
x_l \leq x \leq x_u,
\]

where \( \mu \) is a vector of design objectives, \( g \) and \( h \) are inequality and equality constraint vectors, respectively, \( x_l \) and \( x_u \) are the lower and upper bounds of the design variables, respectively, and \( x \) is a vector of design variables. As stated, Problem 1 yields a particular class of solutions that are said to be Pareto optimal. A design solution \( \mu^* \) is Pareto optimal if there does not exist another solution \( \mu \) such that \( \mu_i \leq \mu_i^* \) for all \( i \in \{1, 2, \ldots, n\} \) and \( \mu_j < \mu_j^* \) for at least one index of \( j \), \( j \in \{1, 2, \ldots, n\} \). Figure 2a depicts the feasible region (shaded) for a multiobjective design problem where the design objectives \( \mu_1 \) and \( \mu_2 \) are minimized. The complete set of Pareto solutions comprises the Pareto frontier, which is shown as a heavy line in Figure 2a.
Associated with every multiobjective optimization problem are important reference points in the multiobjective design space. These reference points are defined below.

**Anchor points** are specific designs, in the feasible design space, that correspond to the best possible values for respective individual objectives. For a bi-objective problem, for example, the anchor points are labeled as $\mu_1^*$ and $\mu_2^*$ in Figure 2a. The $i$th anchor point is written as

$$
\mu_i^* = [\mu_1(x_i^*) \mu_2(x_i^*) \cdot \cdot \cdot \mu_n(x_i^*)]^T,
$$

where $x_i^* = \arg \min_x \mu_i(x)$ subject to the constraints given in Eqs. (2)–(4).

**Utopia point** is a specific point, generally outside of the feasible design space, that corresponds to all objectives simultaneously being at their best possible values. The utopia point is denoted as $\mu^U$ in Figure 2a, and is written as

$$
\mu^U = [\mu_1(x^*) \mu_2(x^*) \cdot \cdot \cdot \mu_n(x^*)]^T.
$$

**Nadir point** is a point in the design space where all objectives are simultaneously at their worst values. The nadir point is generally expressed as

$$
\mu^N = [\mu_1^N \mu_2^N \cdot \cdot \cdot \mu_n^N]^T,
$$

where $\mu_i^N$ is defined as

$$
\mu_i^N = \max_x \mu_i(x),
$$
subject to the constraints given in Eqs. (2)–(4). Another useful way to define $\mu_i^N$ is

$$
\mu_i^N = \max \{ \mu_i(x_1^a), \mu_i(x_2^a), \ldots, \mu_i(x_n^a) \}.
$$

(9)

If Eq. (9) is used to define $\mu_i^N$, then the resulting point is referred to as the pseudo nadir point. The pseudo nadir point is one in the design space with the worst design objective values of the anchor points.

A requisite component of the smart Pareto methodology is the generation of candidate design solutions. The set of candidate designs must adequately represent all regions of the Pareto frontier. This is because the smart filter will not generate new points; it will only judiciously remove points from a provided set. Therefore, it is highly desirable to use a Pareto generation approach that yields Pareto solutions throughout the complete Pareto frontier, so as to not under-represent any one Pareto region. Ideally, the Pareto set generator would also avoid over-representing any one region of the Pareto frontier. To this end, Section 2.2 presents a synopsis of a Pareto-set generator that possesses these two properties.

2.2 A Synopsis of the Normal Constraint Method

The normal constraint method is a Pareto-set generator that guarantees the generation of evenly distributed solutions over the complete Pareto frontier [10, 11]. Under the normal constraint method, the multiobjective optimization problem is converted to a single objective optimization problem that is solved repeatedly, subject to a judiciously constructed set of constraints. After performing a series of optimizations, a set of evenly distributed Pareto solutions results. This section provides a brief description of the normal constraint method; the interested reader may refer to Messac and Mattson [11] for a more detailed presentation. For clarity of presentation, the synopsis begins with a graphical description of the normal constraint method applied to a bi-objective case, followed by an analytical development for the $n$-objective case.

2.2.1 Graphical Description of the Normal Constraint Method

Figures 2a–c are used to illustrate the basic normal constraint method for a bi-objective problem. As discussed earlier, Figure 2a shows the feasible design space (shaded) for the minimization of design objectives $\mu_1$ and $\mu_2$. The Pareto frontier is shown as a heavy curve. Under the normal constraint method, the objectives are normalized (Eq. (11)) and the optimization is performed in the normalized space. The normalization is performed to eliminate any difficulties related to disparate objectives magnitudes. Let $\bar{\mu}$ be the normalized form of $\mu$.

The normalized design space is shown in Figure 2b (this is the normalized form of Fig. 2a). The mapping to the normalized space is provided in Eq. (11).

In the normalized space, a series of optimizations is performed to obtain a set of Pareto solutions. For each optimization run, the feasible region is reduced according to the following approach. A so-called utopia line is constructed between the anchor points. Another line is constructed that is normal to the utopia line. The latter is called the Line NU (‘NU’ indicates that it is normal to the utopia line). The feasible region is then reduced to the area northwest of the Line NU. The objective $\bar{\mu}_2$ is then minimized subject to the reduced feasible space (Fig. 2b), yielding a single Pareto solution, $\bar{\mu}$. It is important to note that the feasible space is reduced to the region northwest of the Line NU. This is a key difference been the normal constraint method and the normal boundary intersection method. The latter requires that solutions be on the normal line. The Line NU is shifted, using even increments, from $\bar{\mu}_1^a$ to $\bar{\mu}_2^a$ along the utopia line. At each shift, an optimization subject to the reduced feasible space yields a Pareto solution. As the Line NU is shifted at even increments, a remarkably well distributed set of
solutions is obtained along the normalized Pareto frontier. The design space is then mapped back to the non-normalized space as shown in Figure 2c.

Note that when there is a difference in objective scaling, the even distribution of Pareto solutions (in normalized space) is stretched when it is mapped back to the non-normalized space. Fortunately, this is of little practical consequence in terms of the physical meaning of the sparsely distributed points. All objectives are well represented, albeit not necessarily by evenly distributed solutions in the non-normalized space.

2.2.2 Mathematical Implementation for Problems with n Objectives

A seven-step process is now presented for implementing the normal constraint method for problems of n objectives.

Step 1 – Obtain anchor points: Obtain the anchor point, $\mu^i$ for all $i \in \{1, 2, \ldots, n\}$.

Step 2 – Normalize objective values: Normalize the design objectives according to the following approach. Define the vector $L$ (Fig. 2a) as

$$L = [l_1, l_2, \ldots, l_n]^T = \mu^N - \mu^U,$$

(10)

where $\mu^N$ is defined by Eq. (9). Normalize the design metrics as

$$\bar{\mu}_i = \frac{\mu_i - \mu_i(x^i)}{l_i} \quad i \in \{1, 2, \ldots, n\}$$

(11)

The normalization process is an important part of the normal constraint method. Without normalization, certain important regions of the Pareto frontier would be under-represented in the Pareto set, for problems having disparate objectives magnitudes.

Step 3 – Define utopia plane vectors: For any given $j \in \{1, \ldots, n\}$, define $n - 1$ vectors as

$$\bar{v}_i = \bar{\mu}_j^* - \bar{\mu}_i \quad \forall i \in \{1, \ldots, n\}; \quad i \neq j.$$

(12)

These vectors are analogous to the utopia line for the bi-objective case.

Step 4 – Compute normalized increments: Compute a normalized increment, $\delta^j$, along the vector $\bar{v}_j$ for a prescribed number of solutions, $m_j$, along the associated $\bar{v}_j$ vector as

$$\delta^j = \frac{1}{m_j - 1} \quad j \in \{1, 2, \ldots, n - 1\}.$$

(13)

Step 5 – Generate hyperplane points: Evaluate the $i$th point on the utopia hyperplane as

$$\tilde{p}_i = \sum_{j=1}^{n} \alpha_i^j \bar{\mu}_j^*,$$

(14)

where the non-dimensional parameter $\alpha_i^j$ satisfies the conditions

$$0 \leq \alpha_i^j \leq 1$$

(15)

and

$$\sum_{j=1}^{n} \alpha_i^j = 1.$$
By varying $\alpha_j$ from 0 to 1 with a fixed increment of $\delta_j$, an even distribution of points on the utopia plane can be generated. Importantly, in order to guarantee complete coverage of the Pareto frontier (for problems of $n$ objectives), the condition imposed by Eq. (15) must be modified according to the detailed developments presented in Messac and Mattson [11]. In short, the upper and lower limits on $\alpha_j$ must be relaxed in order to guarantee complete coverage of the Pareto frontier. It is noted, however, that using Eq. (15) as presented above, strikes a pragmatic balance between computational cost and Pareto frontier coverage.

**Step 6 – Generate Pareto points:** Generate a set of well-distributed Pareto solutions in the normalized objective space. For each value of $\bar{\mu}_k$ generated in Step 5, a corresponding Pareto solution is obtained by solving Problem 2.

**Problem 2: normal constraint generic optimization problem for point $\bar{\mu}_k$**

$$\min_{x} \bar{\mu}_j(x)$$

subject to

$$g(x) \leq 0,$$  \hspace{1cm} (18)

$$h(x) = 0,$$  \hspace{1cm} (19)

$$x_l \leq x \leq x_u,$$  \hspace{1cm} (20)

$$\vec{v}_i \cdot (\bar{\mu} - \bar{\mu}_k) \leq 0; \quad \forall i \in \{1, \ldots, n\}, \quad i \neq j,$$  \hspace{1cm} (21)

$$\vec{v}_i = \bar{\mu}^{j*} - \bar{\mu}^{i*}; \quad \forall i \in \{1, \ldots, n\}, \quad i \neq j,$$  \hspace{1cm} (22)

where $j \in \{1, \ldots, n\}$. Solving Problem 2 for a set of evenly distributed points $\bar{\mu}_k$ will result in an evenly distributed set of Pareto solutions in the normalized space.

**Step 7 – Return to the non-normalized space:** Map the design metric values for the Pareto solutions, obtained in Step 6, back to the non-normalized space using the relationship provided in Eq. (11).

In some cases, the normal constraint method will generate locally Pareto or non-Pareto points. To ensure that the set of generated points is globally Pareto optimal, a filter is developed in the next section that removes all spurious points.

### 2.3 A Global Pareto Filter

In this section, we develop a global Pareto filter that effectively compares a set of candidate designs. The algorithm filters out all non-Pareto solutions and all locally Pareto solutions that may have been obtained during the generation of candidate designs. This section presents a general description of the global Pareto filter. In addition to this general description, a detailed flowchart of the algorithm is also provided (Fig. 3).

The global Pareto filter successively and exhaustively compares generic design pairs $\mu^i$ and $\mu^j$ from a given set of candidate solutions, and removes any solution that is not Pareto optimal. Recall that the mathematical definition of Pareto optimality was given in Section 2.1. As shown in Figure 4, we start with a set of candidate designs; each of which is denoted as $\mu^k$ where $k \in \{1, 2, \ldots, n_p\}$. We then compare design $\mu^i$ with successive $\mu^j$ designs. If during the comparison, it is determined that $\mu^i$ dominates $\mu^j$, then $\mu^j$ is removed from the set of candidate designs (i.e. $\mu^j$ is put in the group of removed designs). The process continues by comparing $\mu^i$ with the next $\mu^j$. Likewise, if $\mu^j$ dominates $\mu^i$, then $\mu^i$ is removed from the candidate set, and the index $i$ is incremented. When neither design dominates, then neither is removed;
both designs are retained in the set of candidate designs. When $\mu^i$ has been exhaustively compared and still remains in the candidate set, then it is globally Pareto optimal; $\mu^i$ joins the set of globally Pareto designs. After the index $i$ has reached its maximum, the set of candidate designs is depleted, and two sets remain: a set of removed designs, and a set of globally Pareto designs.
It is important to note that the final set of designs is globally Pareto only with respect to the initial set of candidate designs. Obtaining the global Pareto set is a critical and integral part of the smart Pareto approach. The reason for this is made evident in Section 2.4. Given the global Pareto set obtained in this section, we are now ready to reduce it to the smart set through the smart Pareto filter presented in Section 2.4.

### 2.4 The Smart Pareto Filter

This filter compares a set of globally Pareto designs and removes all solutions that the designer deems less useful in terms of characterizing! The design tradeoffs. The removed points are such that the tradeoffs they represent are of negligible practical value. The resulting set of Pareto solutions is said to be a smart Pareto set, which is one that is small and that effectively represents the tradeoff properties of the Pareto frontier.

The smart Pareto filter is based on the notion that certain regions of the Pareto frontier may be deemed less useful than others. Although no Pareto solution is objectively better than another Pareto solution, the designer may indeed deem some Pareto solutions more desirable than others. From a practical decision-making perspective, it is useful to remove any Pareto solution that will not be selected as the final design. Given designer specified parameters, the smart Pareto filter removes such points.

Figure 5 illustrates the basic smart Pareto-filter process; for completeness, a figure illustrating the detailed smart Pareto-filter algorithm is also provided (Fig. 7). As shown in Figure 5, the smart Pareto filter begins with a set of global Pareto designs (or points). Through successive comparisons, this set will be depleted, and two sets will be populated: (1) a set of removed Pareto designs, and (2) a set of smart Pareto designs. Generally speaking, we successively compare two generic designs, \( \mu_i \) and \( \mu_j \), from a given set of globally Pareto solutions, and remove any designs within a so-called region of practically insignificant tradeoff (PIT) (Fig. 6). A detailed definition of the PIT region is discussed shortly.

Specifically, we start with a design \( \mu_i \) and compare it with successive \( \mu_j \) designs. When, during this comparison, it is determined that the given design \( \mu_j \) is within the PIT region of \( \mu_i \), then \( \mu_j \) is removed from the set of candidate designs (i.e. \( \mu_j \) is put in the group of removed points, or the PIT points). The process continues as \( \mu_i \) is compared with the next design \( \mu_j \). When the given design \( \mu_j \) is not within the PIT region of design \( \mu_i \), then it is retained, and we proceed to the next \( \mu_j \) that is neither in the removed set nor in the smart set. The process continues by proceeding to the next remaining design \( \mu_i \) in the set under consideration.

![Figure 5 General smart Pareto-filter approach.](image)
declaring it a smart Pareto point, and performing the same series of comparisons with the remaining set of $\mu^j$ designs.

After the index $i$ has reached its maximum, the set of candidate designs is depleted, and two sets remain: a set of removed designs, and a set of smart Pareto designs. According to this algorithm, no single comparison of two given designs is ever repeated. Additionally, once a solution is put in either the removed set or the Pareto set, it is removed from all further comparisons.

As a fundamental part of the smart Pareto filter, the designer declares what he or she deems to be an insignificant tradeoff. This is done by defining a PIT region. To introduce the notion of a PIT region, first consider the bi-objective case illustrated in Figure 6a. As shown in the figure, the Pareto point $\mu^i$ is located at the origin of the axes $\tilde{\mu}_1$ and $\tilde{\mu}_2$, which simply represent deviations of the respective objectives from the point $\mu^i$. Assume that $\mu^i$ has been previously declared to belong to the smart Pareto set. The shaded region constructed near the point $\mu^i$ represents the PIT region. Every Pareto point that lies within the PIT region is said to entail a practically insignificant tradeoff when compared to point $\mu^i$. For problems of three objectives, the PIT volume is shown in Figure 6b. The smart Pareto point to be compared with others in this case is denoted by $\mu^i$. Any other point that is within the volume constructed around $\mu^i$, is said to entail insignificant tradeoff when compared to $\mu^i$.

The designer specifies the PIT region by prescribing two control parameters $\Delta^t$ and $\Delta^r$. By prescribing values for these two control parameters, the designer implicitly makes two dependent statements: (1) Do not keep two Pareto points for which the difference between two corresponding objectives is less than $\Delta^t$. This difference is considered to entail practically insignificant tradeoff. (2) Given that the difference between the corresponding objectives of two Pareto points is less than $\Delta^t$, if the difference between any other corresponding objectives is greater than $\Delta^r$, then both can be retained in the smart Pareto set.

In the smart filtering process, the absolute value of the vector difference between the two points being compared is

$$v = |\mu^j - \mu^i|.$$  \hspace{1cm} (23)

The minimum and maximum components of $v$ are denoted by $v_l$ and $v_u$, respectively. Mathematically, $\mu^j$ is removed when compared to $\mu^i$ if the following conditions are satisfied.

$$v_l < \Delta^t \quad \text{and} \quad v_u < \Delta^r$$  \hspace{1cm} (24)

where $\Delta^t$ and $\Delta^r$ are shown in Figure 6 and are prescribed by the designer.
The parameter $\Delta^t$ is primarily a design objective tradeoff parameter, which is used to remove points that are insignificantly different from others with respect to a given objective. The parameter $\Delta^r$ is primarily a distribution/representation parameter, which is used to control the degree to which flat regions of the Pareto frontier are represented in the smart Pareto set. For each of these parameters, their values are generally different for each design objective, especially when large differences in magnitude exist between objectives. For simplicity, the control parameters are shown to be equal for each objective in Figure 6. In practice, a designer may specify $\Delta^t_i$ and $\Delta^r_i$, which are the control parameters associated with objective $i$.

Figure 2d graphically depicts the process of filtering the global Pareto set to obtain the smart Pareto set (shown as solid points). Consider the Pareto frontier shown in Figure 2d. Begin with the right most Pareto point, which is chosen as an initial point and is declared a smart Pareto point (as a guideline, one of the anchor points may be chosen as the initial point). The PIT regions are constructed for the smart Pareto point (see ‘1’ in Fig. 2d). All Pareto solutions inside the PIT region are then eliminated. The closest neighboring global Pareto point is then declared a smart Pareto point, and PIT regions are constructed for that point. Any point within the newly constructed PIT region is removed. The process repeats itself until all points in the remaining set are smart Pareto points.

![FIGURE 7 Smart Pareto-filter algorithm.](image-url)
SMART PARETO FILTER

The hollow circles illustrated in Figure 2d are Pareto points that have been eliminated by the smart Pareto filter. The solid (filled) circles constitute the smart representation of the Pareto frontier. Five smart Pareto points and their associated PIT regions are shown in Figure 2d. When the process is complete, the remaining smaller set of points meaningfully characterizes the tradeoff properties of the design. The points removed are such that the tradeoffs they represent are of negligible practical value. It is further understood that pertinent local design tradeoffs are sufficiently well represented by the retained/smart neighboring points. The smart Pareto filter, in essence, yields a smaller set of Pareto points that adequately represents the practical design tradeoffs throughout the design space.

The main elements of the smart Pareto methodology have been presented in Section 2. Specifically, we have shown that one may obtain an even distribution of global Pareto solutions by using the normal constraint method in conjunction with the global Pareto filter. Finally, less useful solutions in the global Pareto set are removed by using the smart Pareto filter. The reduced set represents a minimal number of solutions needed to characterize the tradeoff properties of the design. Importantly, this methodology facilitates the critical task of selecting the final design from among a set of Pareto points. It does so by limiting the final set of candidate designs to a small number of options – all of which are significantly different in terms of their performance. Section 3 explores three examples that illustrate the usefulness of the smart Pareto methodology.

3 EXAMPLES

The first example is a bi-objective mathematical example. The second is a tri-objective mathematical example. The final example is a simple truss design problem.

3.1 Example 1: Numerical Bi-objective Problem

\[ \min_{\mu} [\mu_1 \mu_2]^T \]  
\[ \text{subject to} \]
\[ \mu_1 = x_1, \]
\[ \mu_2 = x_2, \]
\[ \left( \frac{\mu_1 - 10}{10} \right)^8 + \left( \frac{\mu_2 - 5}{5} \right)^8 - 1 \leq 0, \]
\[ -10 \leq x_1 \leq 10, \]
\[ -10 \leq x_2 \leq 10. \]

Using the normal constraint method, the set of Pareto solutions shown in Figure 8a is obtained. These points are passed through the smart Pareto filter developed in Section 2.4; the filtered solutions are shown in Figure 8b.

Figures 8a and b show that the Pareto set is reduced from 21 points to 7 points. Both figures represent the same frontier, but the frontier in Figure 8b is represented with a minimal number of points for given control parameters. Note that the density of Pareto points has been reduced in regions where insignificant tradeoffs occur between the objectives (see the Pareto frontier region where \( \mu_1 \) ranges from 3 to 9, and the region where \( \mu_2 \) ranges from 1 to 5). Thus, the designer now has a smaller set of Pareto solutions from which to choose the final design.
Importantly, this set of points is diverse and comprises only designer-specified useful solutions. Figure 8b was generated using $\Delta^t_i$ and $\Delta^r_i$ equal to 0.15 and $\infty$, respectively, for all $i \in \{1, 2\}$. Considering the objective $\mu_1$, for example, the control parameter $\Delta^t_i$ physically means that any design that is within 0.15 $\mu_1$-units of another design is considered insignificantly different and should be removed from the candidate set. It is important to keep in mind that the designer can explore the impact of changing the control parameters $\Delta^t_i$ and $\Delta^r_i$ in acquiring the desired size of the smart Pareto set.

### 3.2 Example 2: Tri-objective Problem

\[
\begin{align*}
\min_x [\mu_1 \mu_2 \mu_3]^T \\
\text{subject to} \\
\mu_1 &= x_1, \\
\mu_2 &= x_2, \\
\mu_3 &= x_3, \\
(\mu_1 - 1)^4 + (\mu_2 - 1)^4 + (\mu_3 - 1)^4 - 1 &\leq 0.
\end{align*}
\] (31)

The initial set of globally Pareto points is obtained for Example 2 using the normal constraint method (with the pseudo nadir point and $0 \leq \alpha_j \leq 1$) and is shown in Figure 9a. The smart Pareto filter is used to obtain the reduced Pareto set shown in Figure 9b. The results were generated using $\Delta^t_i$ and $\Delta^r_i$ equal to 0.0075 and 0.1, respectively, for all $i \in \{1, 2, 3\}$. The objective space in Example 2 is particularly flat in the areas near points (1, 1, 0), (1, 0, 1), and (0, 1, 1). A comparison of Figure 9a and b (especially in the regions indicated) shows that indeed, the smart Pareto filter reduces the point density in areas of insignificant tradeoff. We make the important comment that the degree of this reduction in density is fully controlled by the parameters $\Delta^t_i$ and $\Delta^r_i$.

### 3.3 Example 3: Truss Design Problem

In this problem, we wish to identify the optimal position of the node $P$ and cross-sectional areas for the left and right bars, such that the nodal displacement at node $P$ and structural volume are minimized, when the loads $W_1$ and $W_2$ are applied as shown in Figure 10.
The optimization problem statement for the truss example is given as

\[
\min_{a, b} \left\{ \frac{\mu_1(a, b)}{\mu_2(a, b)} \right\} = \min_{a, b} \left\{ \frac{\text{Displacement squared}}{\text{Volume}} \right\}
\]

subject to

\[
S_i \leq S_{\text{max}} \quad \{i = 1, 2\},
\]

\[
a_{\text{max}} \leq a_i \leq a_{\text{max}} \quad \{i = 1, 2\},
\]

\[
b_{\text{min}} \leq b \leq b_{\text{max}}.
\]

The stress in each bar must be lower than the maximum allowable stress, as indicated by Eq. (37), where the bar on the left (Fig. 10) is Bar 1 and that on the right is Bar 2. The cross-sectional area of each bar, \(a_i\), is bounded as described by Eq. (38), and the horizontal location of node \(P, b\), is also constrained as shown by Eq. (39). The fixed parameters for this problem are the Young’s modulus, \(E = 200 \text{ GPa} (29 \times 10^3 \text{ ksi})\); the truss height, \(L = 18.3 \text{ m}\).
(60 feet); the maximum allowable stress, $S_{\text{max}} = 3.8 \text{ GPa (550 ksi)}$; and the loads, $W_1$ and $W_2$, are $4.45 \times 10^5 \text{ N (100 kips)}$ and $4.45 \times 10^6 \text{ N (1000 kips)}$, respectively. Additionally, $a_{\text{min}} = 5.16 \times 10^{-4} \text{ m}^3 (0.8 \text{ in}^3)$ for $\{i = 1, 2\}$; $a_{\text{max}} = 1.94 \times 10^{-3} \text{ m}^2 (3.0 \text{ in}^2)$ for $\{i = 1, 2\}$; $b_{\text{min}} = 9.144 \text{ m (30 ft)}$, and $b_{\text{max}} = 27.432 \text{ m (90 ft)}$.

Example 3 is solved using the normal constraint method; the results of which are shown in Figure 11a. The smart Pareto filter is used to obtain the reduced Pareto set as seen in Figure 11b, where $\Delta_1^i$ and $\Delta_2^i$ equal 0.008 and 0.2, respectively, for all $i \in \{1, 2\}$. Table 1 provides numerical results for the smart Pareto points of Figure 11b. Figure 11b also shows that many designs in the range of displacement equalling 0.2–0.5 m² do not entail significant tradeoff in terms of volume, and thus are eliminated by the smart filter.

![Figure 11](image.png)

**TABLE I** Results for two bar truss example.

<table>
<thead>
<tr>
<th>Design</th>
<th>Displacement (m²)</th>
<th>Volume (m³)</th>
<th>$a_1$ (m²)</th>
<th>$a_2$ (m²)</th>
<th>$b$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.48</td>
<td>0.04</td>
<td>0.00102</td>
<td>0.00052</td>
<td>11.25</td>
</tr>
<tr>
<td>2</td>
<td>0.32</td>
<td>0.04</td>
<td>0.00105</td>
<td>0.00057</td>
<td>9.14</td>
</tr>
<tr>
<td>3</td>
<td>0.22</td>
<td>0.05</td>
<td>0.00117</td>
<td>0.00075</td>
<td>9.14</td>
</tr>
<tr>
<td>4</td>
<td>0.16</td>
<td>0.06</td>
<td>0.00137</td>
<td>0.00088</td>
<td>9.14</td>
</tr>
<tr>
<td>5</td>
<td>0.12</td>
<td>0.07</td>
<td>0.00161</td>
<td>0.00103</td>
<td>9.14</td>
</tr>
<tr>
<td>6</td>
<td>0.09</td>
<td>0.08</td>
<td>0.00186</td>
<td>0.00118</td>
<td>9.14</td>
</tr>
<tr>
<td>7</td>
<td>0.07</td>
<td>0.09</td>
<td>0.00194</td>
<td>0.00139</td>
<td>9.14</td>
</tr>
<tr>
<td>8</td>
<td>0.06</td>
<td>0.10</td>
<td>0.00194</td>
<td>0.00173</td>
<td>9.14</td>
</tr>
</tbody>
</table>
These examples have shown various results obtained by applying a smart Pareto filter. As expected in each case, the Pareto set is reduced by using the smart Pareto filter in a way that retains higher concentration in regions of practical significance. Example 2 is provided primarily to illustrate the use of the smart Pareto filter for problems of $n$ objectives.

4 CONCLUDING REMARKS

This paper presented a novel approach to obtain a so-called smart Pareto set. The smart Pareto set is one that is adequately small, represents the complete Pareto frontier, emphasizes the regions of the Pareto frontier that entail significant tradeoff, and de-emphasizes the regions corresponding to little tradeoff. A smart Pareto methodology was presented and described as the main body of the paper. The methodology included (i) generating a set of candidate optimal solutions, (ii) filtering the set of candidate optimal solutions to remove all that were not globally Pareto, and (iii) filtering the global Pareto set to remove Pareto solutions that were deemed unnecessary by the designer for tradeoff representation purposes. Ultimately, the smart Pareto methodology results in a smaller set of useful Pareto solutions. This small set of Pareto solutions circumvents potential difficulties associated with comparing a significantly large number of Pareto designs in order to choose a final one.

NOMENCLATURE

\( \tilde{()} \) Normalized form of variable
\( \bar{()}_l \) Lower bound of variable
\( \bar{()}_u \) Upper bound of variable
\( \alpha \) Scalar that defines the position of \( \bar{p}_i \)
\( \Delta^{l}, \Delta^{t} \) Control parameters for the smart Pareto filter
\( \delta \) Normalized increment for shifting the normal constraint
\( \bar{g} \) Inequality constraint vector
\( \bar{h} \) Equality constraint vector
\( \bar{L} \) Vector of lengths used to normalize the design objective space
\( m_i \) Number of points along the vector \( \bar{v}_i \)
\( \bar{\mu} \) Design objectives vector (or design metrics vector)
\( \mu^{i*} \) \( i \)th anchor point
\( \mu^{U} \) Utopia point
\( \mu^{N} \) Nadir point
\( \bar{\mu} \) Deviation of \( \bar{\mu} \) from the point \( \mu^{i} \)
\( n \) Number of design objectives
\( n_p \) Number of points in the Pareto set
\( n_x \) Number of design variables
\( \bar{p}_i \) Generic point on the utopia plane
\( \bar{v}_i \) \( i \)th vector defining the utopia plane
\( x \) Vector of design variables
\( x^{i*} \) Optimal design variable values corresponding to the \( i \)th anchor point

Acknowledgements

This research was supported by National Science Foundation Grant DMI-0196243.
References


