2012-05-09

A Method for Exploring Optimization Formulation Space in Conceptual Design

Shane Keawe Curtis
Brigham Young University - Provo

A Method for Exploring Optimization Formulation Space

in Conceptual Design

Shane K. Curtis

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Christopher A. Mattson, Chair
Spencer P. Magleby
C. Greg Jensen

Department of Mechanical Engineering

Brigham Young University

June 2012

ABSTRACT

A Method for Exploring Optimization Formulation Space
in Conceptual Design

Shane K. Curtis
Department of Mechanical Engineering, BYU
Master of Science

Formulation space exploration is a new strategy for multiobjective optimization that facilitates both divergent searching and convergent optimization during the early stages of design. The formulation space is the union of all variable and design objective spaces identified by the designer as being valid and pragmatic problem formulations. By extending a computational search into the formulation space, the solution to an optimization problem is no longer predefined by any single problem formulation, as it is with traditional optimization methods. Instead, a designer is free to change, modify, and update design objectives, variables, and constraints and explore design alternatives without requiring a concrete understanding of the design problem a priori. To facilitate this process, a new vector/matrix-based definition for multiobjective optimization problems is introduced, which is dynamic in nature and easily modified. Additionally, a set of exploration metrics is developed to help guide designers while exploring the formulation space. Finally, several examples are presented to illustrate the use of this new, dynamic approach to multiobjective optimization.

Keywords: design space exploration, multiobjective optimization, conceptual design

ACKNOWLEDGMENTS

First, I would like to thank my graduate committee chair, Dr. Mattson, for his tutelage during my undergraduate and graduate research career at BYU. He has helped me to grow and develop both as a researcher and as a person. I am also grateful to the other members on my graduate committee, Dr. Jensen and Dr. Magleby, for the feedback that they have given me on this research. I express gratitude to the Department of Mechanical Engineering at BYU for the use of their facilities. Importantly, this research was partially supported by National Science Foundation Grant 0954580.

I would like to thank the other students in the Design Exploration Research Group, as their insights and ideas helped me form and refine the theories found in this thesis. In particular, Braden Hancock, Patrick Lewis, Nick Wasley, and Morgan Tackett have made significant contributions to this research. Additionally, I want to recognize the students in the 2011 BYU ME 576 class for their work in developing the physical prototypes found in the second case study of this thesis: Morgan Tackett, Brad Smith, Allan Tribe, Prasad Weerakoon, Garrett Bodily, Clark Davis, Sam Wilding, Brad Solomon, Robert Campbell, and David Olligschlager. Finally, I am eternally grateful to my wife, Sam, for all the love and support that she has shown me, despite my shortcomings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$\mathbf{g}$	Vector of inequality constraints
$\mathbf{h}$	Vector of equality constraints
$\mathbf{p}$	Vector of fixed design parameters
$\mathbf{v}$	Vector of interest in the objective space
$\mathbf{w}$	Diagonal matrix of objective weights
$\mathbf{x}$	Vector of design variables or design objects
$\mathbf{y}$	Vector of independent design objects
$\mathbf{z}$	Vector of dependent design objects
μ	Vector of design objectives
γ	Vector in the objective space

Subscripts and Subscripts

$[\,]_i$	Dummy index
$[\,]_r$	Dummy index
$[\,]_s$	Dummy index
$[\,]_l$	Lower bound
$[\,]_u$	Upper bound
$[\,]^{(-)}$	Lower bound in inverse optimization
$[\,]^{(+)}$	Upper bound in inverse optimization
$[\,]^{(0)}$	Indicates benchmark
$[\,]^{(k)}$	Indicates a concept or formulation
$[\,]^U$	Utopia
$[\,]^N$	Nadir
$\underline{[\,]}$	Indicates the formulation space

CHAPTER 1. INTRODUCTION

Advancements in computational power have transformed the way engineers perform product design, especially during the later, detailed stages of the design process. Computer-aided design software, finite element analysis, computational fluid dynamics, and numerical optimization are just a few of the computational tools at the designer's disposal. However, many of these tools, especially numerical optimization, are rarely utilized during early, conceptual design stages. There are many reasons for this, but most stem from the nature of conceptual design, which is typically qualitative and fluid in nature. Numerical optimization, on the other hand, usually requires quantitative, well-defined problems to solve. Thus, there is a disconnect between numerical optimization and conceptual design. The objective of this thesis is to provide a numerical optimization method that is well-suited for use during conceptual design, allowing designers to benefit from the power of computational assistance and make more informed decisions earlier in the design process when their impact is the greatest.

Conceptual design has been defined in various ways by several researchers [1–3]. A common thread among all these definitions is the generation and discovery of design requirements/possibilities, coupled with the analysis and selection of design concepts for further development. Prevalent activities during conceptual design include benchmarking, conducting market research, abstracting the problem, sketching new ideas, brainstorming, building rudimentary prototypes, developing low fidelity analytical models, and testing design concepts by simple experimentation. Clearly, if computational assistance is to be utilized during conceptual design, then an analytical model in some form is needed. Therefore, in the context of this thesis, a design concept is defined as an idea that has evolved to the point where there is a parametric model that represents one or more aspects of its performance [4].

Consider a hypothetical, conceptual design process depicted in Fig. 1.1. The top portion of the figure represents the designer's actions, while the bottom represents the computer's. Impor-

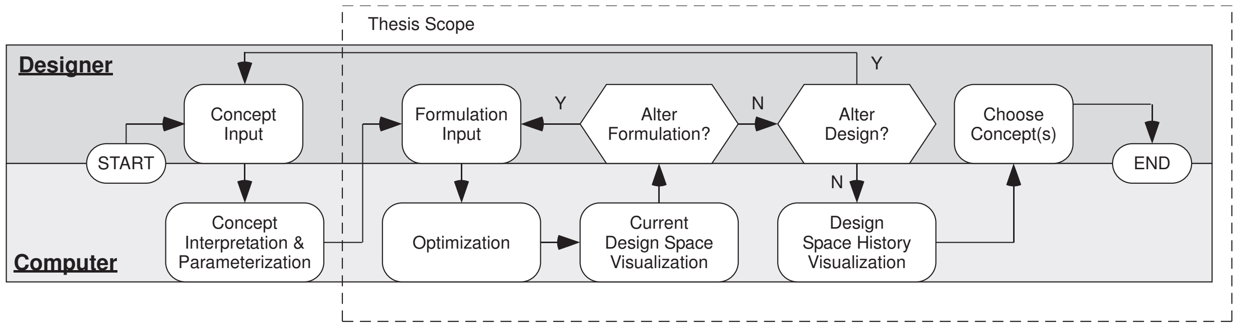


Figure 1.1: A synergistic design exploration process is displayed. The boxes on the top indicate human activities, while the boxes on the bottom are performed by the computer.

tantly, this structure is designed to be synergistic, placing the designer in a position of creativity and exploration, while positioning the computer for computational searching and mundane calculations. The first step is for the designer to enter a concept input – a sketch for example. The input is then interpreted and parameterized by the computer. After the computer has parameterized the design input, the design parameters are presented to the designer for use in exploring different optimization problem formulations (i.e., specifying design variables, constraints, and objectives). The ensuing iterative procedure of optimizing, viewing the results, and reformulating the optimization problem is called *formulation space exploration* and is more formally introduced in Chapter 4 of this thesis. In this manner, the designer is able to generate, analyze, explore, and select design concepts with the aid of computational assistance.

While the process in Fig. 1.1 is hypothetical, the ideas presented therein have the potential to enhance existing conceptual design procedures. The purpose of this thesis is to develop the supporting theories and methods to carry out the steps inside the dashed box. Specifically, the goal here is to provide an optimization framework that (i) allows engineers to evolve and search a design space efficiently and interactively during the divergent, exploratory phases of product design, and (ii) adapts easily to new design requirements as they are discovered.

CHAPTER 2. LITERATURE REVIEW

Success in engineering design is closely tied to a designer's ability to make rational, informed decisions throughout the product development process. Decisions that typically have the largest impact on a design's outcome occur during early conceptual design when the least is known about design objectives or constraints [5–8]. While many ad-hoc, heuristic methods [1, 2, 9, 10] exist to support conceptual design decision making, computational search methods are rarely utilized until the later, detailed design stages. Thus, to a large extent, the valuable information that is provided through computational search (i.e., numerical optimization) is only available when its impact on the design's outcome is less noticeable.

Accordingly, several researchers have begun to address the difficulties of implementing numerical optimization techniques during the early stages of product development. For example, to capture and represent qualitative objectives, researchers have turned to interactive genetic algorithms [11–14], fuzzy logic systems [15, 16], and preference based modeling [17]. Multiobjective optimization strategies have been applied to concept generation [18] and concept selection [4, 19], both of which are common conceptual design activities. Several engineering applications where multidisciplinary design optimization has been applied during conceptual design include aircraft configurations [20], communication satellites [21], multistage space launch vehicles [22], and gas turbine engines [23]. Thus, there is significant promise that numerical optimization can be utilized to an even greater extent during conceptual design. Despite these advances, there are still challenges that currently limit the extent to which designers can use computational search methods to assist in early-stage design decision making.

One challenge is that much of early-stage, conceptual design is qualitative in nature. Designers use sketches and rough prototypes to explore concept ideas; very little quantitative modeling takes place. To help bridge this gap, some have used sketch recognition software to transform hand-drawn sketches into parametric, computational models [24–28]. When analytical models do exist, they are often used to explore a large design space; therefore, computationally inexpensive models are advantageous because they allow the designer to quickly explore many design alternatives [29, 30]. To this end, the use of Fourier series [31] and metamodeling/surrogate modeling techniques [32, 33] have been used to obtain adequate analytical models for use during conceptual design exploration and optimization. Of course, decreasing the model's fidelity will inevitably introduce uncertainty; fortunately, methods exist for assessing the uncertainty in surrogate models [34, 35].

Uncertainty is pervasive in design and is caused by variations in consumer perception, available market data, material properties, manufacturing precision. As such, the effects of uncertainty can and should significantly affect decision making in engineering design. Within the literature, two broad categories of non-deterministic approaches to determining levels of uncertainty are identified: *reliability-based design methods* [36–39] and *robust design based methods* [40–46]. Reliability-based design optimization focuses on assessing the probability of design failure and reduces such probabilities by shifting the mean performance away from constraint limits [39]. Robust design optimization focuses on optimizing the mean performance and minimizing performance variation, while maintaining feasibility with probabilistic constraints [44, 47, 48].

While quantifying uncertainty is important to effective decision making [49], it is also critical to allow for *adaptation* due to uncertainty. This is especially important during conceptual design due to its dynamic nature—design parameters, variables, constraints, objectives, and limits are likely to change and evolve over time as the designer learns more about the design problem. With traditional optimization, one must know and clearly define the design parameters, variables, constraints, and objectives before optimization can begin [50]; however, when the optimization problem definition is improperly formulated (i.e., objectives and constraints are erroneously assumed), the designer will likely be unsatisfied with the results [51, 52]. Therefore, in a conceptual design environment where constraints and objectives are unclear, the probability of a designer improperly formulating the optimization problem increases, and as a result this decreases designer confidence in the optimization results. In order to be more effective in conceptual design, the optimization problem formulation should be dynamic in nature—easy to formulate, reformulate, and allow for expansion into regions beyond the space defined by the initial parameterization [53].

Exploration into regions that are not initially defined as part of the search space is called divergent exploration. The purpose of divergent exploration during conceptual design is to discover a rich set of novel design alternatives, from which a solution can be selected for further development during detailed design [54, 55]. Without fully exploring the design space in this manner, it is difficult, if not impossible, to know if a designer has truly found an optimal design. One of the major disadvantages with traditional product development processes is that they tend to converge quickly, and not necessarily to a suitable solution space. If a poor solution is initially chosen, then time and money are wasted during costly design iterations [56]. Thus, it is in the designer's best interest to diverge and explore the design space early in design.

Creativity plays an important role in design space exploration. An abundance of research has been performed in the area of design creativity to determine the factors that facilitate or hinder innovation [57–60]. If a designer understands the do's and don'ts of creativity, they can avoid idea affixation and explore the design space more effectively. This is especially important in early design when it is beneficial to generate as many potential design solutions as possible. While traditional ideation methods, such as brainstorming [61], morphology [62], and synectics [63], have been around for decades and are very useful, they do not capitalize on computation power. If partnered synergistically, humans and computers have a great potential to enhance design creativity [64, 65]. On the other hand, traditional computer design tools, such as computer-aided design (CAD), finite element analysis (FEA), and computational fluid dynamics (CFD) software, may actually hinder the creativity of the designer because they require detailed information and can therefore have *clumsy* interfaces [66]. In fact, it has been shown that premature idea fixation may occur if the perceived cost of modifying a CAD model is too expensive [67]. If computational assistance is to be effectively used in conceptual design, the designer must be allowed to remain in an exploratory, creative mind-set. Specific to numerical optimization, the human designer needs to remain in the loop as the rational decision maker. In this sense, optimization is designed to support human decision making—not replace it [68, 69].

The objective of this thesis is to provide a numerical optimization method that is tailored for the early, creative stages of engineering design. As identified in the literature above, the method must (i) allow for adaptation due to the dynamic nature of conceptual design, (ii) allow for both divergent and convergent searches, and (iii) keep the designer in the loop.

The remainder of the thesis is organized as follows: In Chapter 3, the standard multi-objective optimization problem is reviewed. In Chapter 4, the fundamental theory for divergent exploration is introduced, followed by the presentation of a new, dynamic optimization problem formulation in Chapter 5. Exploration metrics to help guide the designer during the exploration process are given in Chapter 6. Three use scenarios for formulation space exploration are explained in Chapter 7. The theories developed in this paper are then illustrated with two case studies in Chapter 8, and concluding remarks are offered in Chapter 9.

CHAPTER 3. TECHNICAL PRELIMINARIES

The generic deterministic multiobjective optimization problem formulation is typically given as Problem 1 (*P1*):

$$\min_{\mathbf{x}} \{\mu_1(\mathbf{x},\mathbf{p}),\ \mu_2(\mathbf{x},\mathbf{p}),\ \ldots,\ \mu_n(\mathbf{x},\mathbf{p})\} \quad (n \geq 2) \tag{3.1}$$

subject to inequality constraints, equality constraints, and side constraints

$$\mathbf{g}_r(\mathbf{x},\mathbf{p}) \leq 0 \quad \{r = 1,2,...,n_g\} \tag{3.2}$$

$$\mathbf{h}_s(\mathbf{x},\mathbf{p}) = 0 \quad \{s = 1,2,...,n_h\} \tag{3.3}$$

$$x_{l,i} \leq x_i \leq x_{u,i} \quad \{i = 1,...,n_x\} \tag{3.4}$$

where μ_i denotes the i-th generic design objective function; $\mathbf{x}$ is a vector of design variables; $\mathbf{p}$ is a vector of fixed design parameters; and n_g, n_h, and n_x, are the total number of inequality constraints, equality constraints, and design variables, respectively. As a note, μ, $\mathbf{g}$, and $\mathbf{h}$ may be linear or non-linear functions of $\mathbf{x}$ and $\mathbf{p}$.

As formulated above, *P1* yields a set of optimal design alternatives—those belonging to the Pareto frontier. This is shown graphically in Fig. 3.1, where the feasible design objective space for two minimized objectives (μ_1 and μ_2) is plotted. In the figure, any point residing on or in the shaded region represents a feasible design solution, meaning that the inequality, equality, and side constraints for the design are satisfied. Each solution comprising the frontier is said to be *Pareto optimal*, which means there are no other designs for which *all* objectives are better satisfied [70–73]. Designers generally seek Pareto solutions because they indicate that the objectives have been improved as much as possible without sacrificing the performance of another competing objective [70].

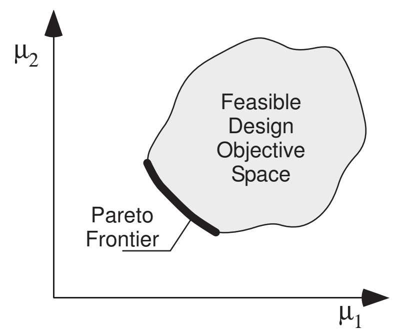


Figure 3.1: Feasible design objective space is shown shaded and the Pareto frontier is shown as the bolded curve.

Other important definitions associated with *P1* include the *utopia point* and the *nadir point*. The utopia point, μ^U, is the point where every objective is simultaneously at its best, or

$$\mu^U = [\mu_1^U, \mu_2^U, ..., \mu_n^U]^T \tag{3.5}$$

where μ_i^U is defined as

$$\mu_i^U = \min_{\mathbf{x}} \ \mu_i(\mathbf{x}, \mathbf{p}) \tag{3.6}$$

subject to the inequality, equality, and side constraints on *P1*. Likewise, the nadir point, μ^N, is the point where every objective is simultaneously at its worst, or

$$\mu^N = [\mu_1^N, \mu_2^N, ..., \mu_n^N]^T \tag{3.7}$$

where μ_i^N is defined as

$$\mu_i^N = \max_{\mathbf{x}} \ \mu_i(\mathbf{x}, \mathbf{p}) \tag{3.8}$$

subject to the same constraints. Typically, neither the utopia or nadir points are on the Pareto frontier nor are they typically realizable; however, they are helpful for characterizing the bounds of an optimization problem search space [74].

Problem 1 is well suited for optimization routines used later in the design process when design objectives and constraints are known and the goal is to converge to the optimal solution. However, in early-stage design, design objectives and constraints may be unknown and the goal is often to diverge and explore many design alternatives, building designer confidence that a better design has not been overlooked [1].

CHAPTER 4. FORMULATION SPACE EXPLORATION

A primary objective of this thesis is to introduce a new optimization strategy that is well-suited for use during conceptual design. The fundamental concept behind this method will be introduced using a simple, yet popular engineering problem: the two-bar truss. The truss is depicted and labeled at the top of Fig. 4.1. The graphs below the truss, labeled (a) through (d), represent a traditional view of the *design space* for the two-bar truss. The graphs in (e) and (f) introduce a new concept in optimization, which is centered on the idea that the design variable space and objective space of a particular formulation represent only a portion of a larger space known as the *formulation space*, the exploration of which is beneficial to the designer.

The two-bar truss at the top of Fig. 4.1 is composed of circular tubing. The solid lines indicate the undeflected state of the truss, while the dashed lines represent the deflected state. The independent design parameters and variables defined in the figure include the height (H), base length (B), tube diameter (d), tube wall thickness (t), material density (ρ), modulus of elasticity (E), and vertical load (P). The mass (M), stress (σ), buckling stress constraint ($\sigma_{buckling}$), and deflection (δ), are calculated using a strength/mechanics of materials model [75]. Using these equations, it is possible to formulate an optimization problem for the truss. For instance, δ and M can be minimized by changing t and H, subject to inequality constraints on σ and $\sigma_{buckling}$.

Figure 4.1(a) represents the design variable space for t and H. Here, each design variable is shown on one of two orthogonal axes for this two-variable problem. Clearly design spaces are not limited in concept to two dimensions. However, this is done here for illustration purposes. The shaded region is the *feasible* design variable space. Any point in the space that resides in this region satisfies all constraints placed on the design variables. The points or designs in the design variable space map to the design objective space through the objective functions. The design objective space for δ and M is shown in Fig. 4.1(b). Again, the set of feasible designs in the design objective space is represented by the shaded region. All solutions in this region satisfy all the constraints of

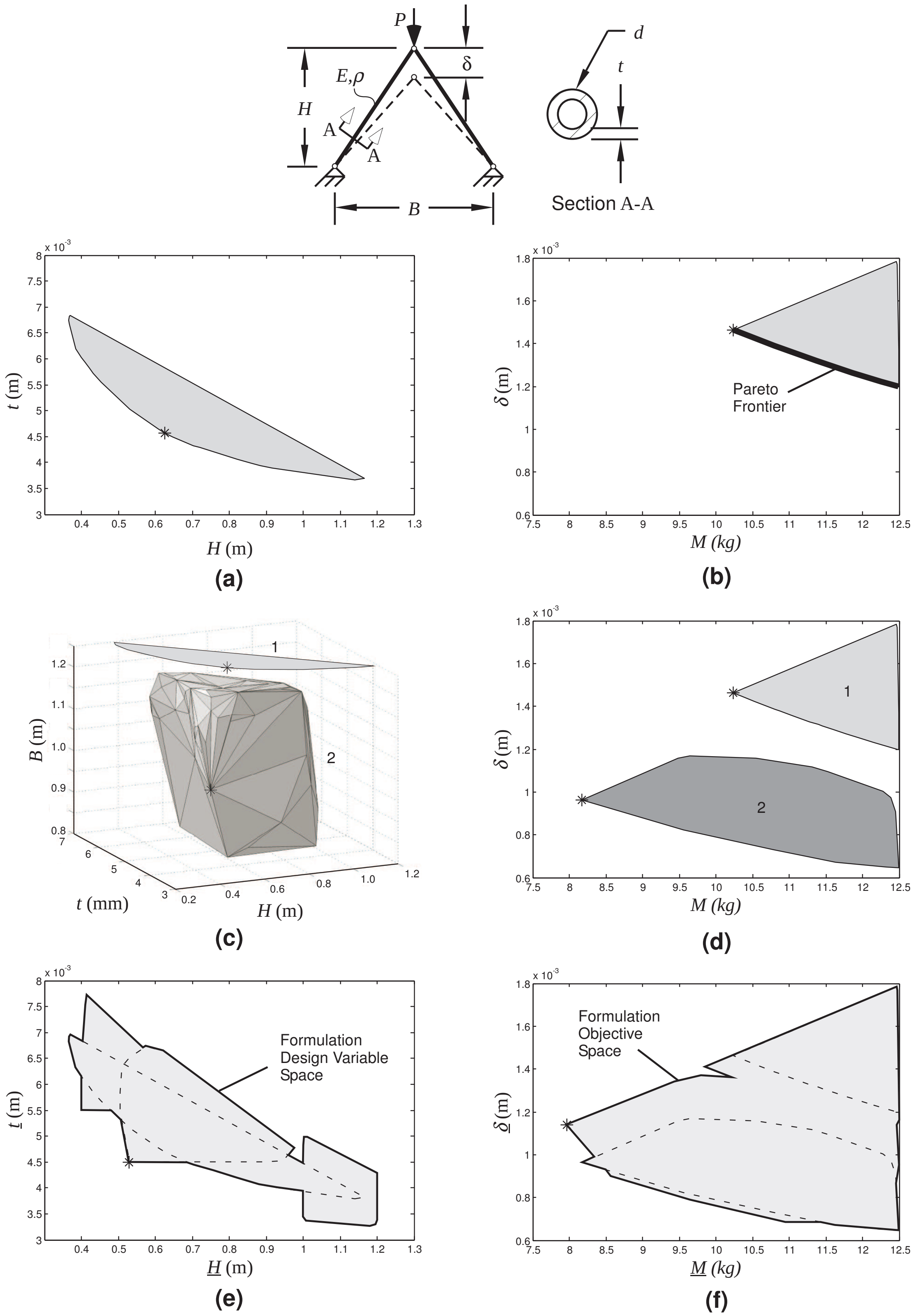


Figure 4.1: Top: two-bar truss (a) Design variable space for Formulation 1 (b) Design objective space for Formulation 1 (c) Design variable space for Formulations 1 and 2 (d) Design objective space for Formulations 1 and 2 (e) Formulation design variable space (f) Formulation design objective space.

the problem formulation. Recall that for two or more objectives, the Pareto frontier (see Chapter 3) exists if two or more objectives are in conflict—which is the case here, since a decrease in material (and therefore, mass) can be expected to result in an increase in the deflection (δ) of the truss.

It is essential to note that both the design variable space and the design objective space are completely defined by the optimization problem formulation. In fact, if the optimization problem is reformulated to include B as a design variable (i.e., minimize δ and M by changing t, H, and B subject to the same inequality constraints on σ and $\sigma_{buckling}$), then different design variable and objective spaces result, which are the darker shaded regions in Fig. 4.1(c) and (d). The new design variable space in (c) is a 3-dimensional space due to the adjusted values of the design variable B. It maps to a completely separate design objective space in (d). The design of least mass, seen as an asterisk in the graphs, is different depending on the optimization formulation. Thus, a common argument against numerical optimization methods is that the optimal solution to the problem is predefined by the problem formulation—in other words, the optimal solution is defined before the search begins. For many practical problems this predefinition is not a drawback since numerical optimization is employed to simply carry out the mundane search for the solution that the designer knows he or she wants. For other design problems, not of this nature, the designer is genuinely interested in exploring the design options *without having to have formed a concrete understanding of the problem or definition of the formulation*. In such cases, which are abundant in early design, an alternative concept for numerical optimization is needed.

The new strategy presented here expands the exploration of design possibilities to another space—the *formulation space*, meaning the optimization problem formulation space. Exploration into this space is divergent, as it expands from the traditional consideration of design variable and design objective spaces for only *one* problem formulation. Figures 4.1(e) and (f) illustrate the formulation space for design variables t and H, and the formulation space for design objectives δ and M, respectively. For notation purposes, a bar is placed under the symbol to indicate that it is in the formulation space. To elaborate, consider Fig. 4.1(e), which is the formulation space for design variables. Shown here is a large, shaded region labeled as the formulation design variable space. The regions enclosed by the dashed lines represent the 2-dimensional projections of the design variable spaces formed by the previous optimization problem formulations from above. As shown, the formulation space encompasses the previous formulations, and expands into design spaces that have not explicitly been introduced here. The formulation space is the union of all design variable and objective spaces identified by the designer as being valid and pragmatic problem formulations.

By expanding the exploration into the formulation space, the optimal solution is no longer predefined by the optimization formulation; instead, the solution is formed through divergent exploration of the formulation space. Importantly, this places the designer at the center of the optimization loop, where his or her judgment can be utilized to rationally interpret the results of the computational search.

Divergence in early design is crucial to avoid missing a potentially superior solution, only to later discover its existence and have to perform costly design iterations [1]. With *P1*, once the problem is formulated, convergence begins and divergence can no longer occur unless modifications are made to the programming of the problem formulation. Changing the formulation in *P1* after it has been executed is non-trivial, as the designer must transition from a creative, explorative mind-set to an analytical mental disposition to reprogram the optimization problem. This is not conducive to design exploration; the designer is less likely to ask "what if" questions if he or she must exert a significant amount of effort to reformulate the optimization problem. In CAD modeling, studies have shown that premature idea fixation is likely to occur if the perceived cost of changing the model is too high (see Chapter 2). The same is true for optimization—if the perceived cost of reformulating the optimization problem is high, then a designer will not likely explore the formulation space. Thus, a dynamic optimization problem formulation is required, one that allows the designer to optimize and reoptimize with modified variables, constraints, or objectives at a low cost.

CHAPTER 5. DYNAMIC MULTIOBJECTIVE OPTIMIZATION PROBLEM

Formulation space exploration requires us to look at optimization problem statements in a new light—one in which design variables can seamlessly turn into design parameters, or inequality constraints into design objectives, etc. As optimization problem formulations change, so do the individual components; what was a design parameter in one formulation could be implemented as a design objective in the next formulation. Thus, to avoid confusion and to illustrate that the designer does not have to commit to variables, parameters, constraints, or objectives, these components of an optimization formulation will be referred to as *design objects*. With this understanding, a generic dynamic multiobjective optimization problem is presented as Problem 2 (*P2*):

$$\min_{\mathbf{y}} \ \{\mu_1(\mathbf{x}),\ \mu_2(\mathbf{x}),\ ...,\ \mu_{n_x}(\mathbf{x})\} \quad (n_x \geq 2) \tag{5.1}$$

subject to the side constraints

$$y_{l,i} \leq y_i \leq y_{u,i} \quad \{i = 1,...,n_y\} \tag{5.2}$$

$$z_{l,i} \leq z_i \leq z_{u,i} \quad \{i = 1,...,n_z\} \tag{5.3}$$

where

$$\mu = \mathbf{w} * \mathbf{x} \tag{5.4}$$

$$\mathbf{w} = \begin{bmatrix} w_{1,1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_{n_x,n_x} \end{bmatrix} \tag{5.5}$$

$$\mathbf{x} = \left[y_1, y_2, ..., y_{n_y}, z(y)_1, z(y)_2, ..., z(y)_{n_z}\right]^T \tag{5.6}$$

Table 5.1: A description of the conditions that dictate design object behavior in *P2*.

Design Object x_i	**Condition**
Minimized Objective	$w_{i,i} > 0$
Maximized Objective	$w_{i,i} < 0$
Non-Objective	$w_{i,i} = 0$
Design Parameter	$y_{l,i} = y_{u,i}$
Design Variable	$y_{l,i} \neq y_{u,i}$
Equality Constraint	$z_{l,i} = z_{u,i}$
Inequality Constraint	$z_{l,i} \neq z_{u,i}$

and **x** is a vector composed of independent design objects (model inputs), **y**, and dependent design objects (model outputs), **z**; **w** is a diagonal matrix where each element along the diagonal is bound between -1 and 1 (i.e., $-1 \leq w_{i,i} \leq 1$); n_x, n_y, n_z are the number of design objects, independent design objects, and dependent design objects, respectively.

Problem 2 is very similar to Problem 1, with a few exceptions. Beginning with Eq. 5.1 in *P2*, the dynamic multiobjective optimization problem is minimized over all independent design objects in **y** instead of only the design variables in **x**. In fact, assume the nature of **x** has changed; it now includes all independent and dependent design objects, whereas in *P1*, **x** only contained independent design variables. The role of each design object in **x** is determined by the lower and upper bounds on **y** and **z** in Eqs. 5.2 and 5.3, as well as the values in the diagonal of **w** in Eq. 5.5. If in Eq. 5.2, $y_{l,i} = y_{u,i}$, then y_i (or x_i) is a design parameter; otherwise, y_i is a design variable. Likewise, if in Eq. 5.3, $z_{l,i} = z_{u,i}$, then z_i (or x_{i+n_y}) is an equality constraint; otherwise, z_i is an inequality constraint. Thus, the inequality, equality, and side constraints for *P1* are satisfied with Eqs. 5.2 and 5.3. If in Eq. 5.5 $w_{i,i} = 0$, then the corresponding x_i is not a design objective. If $w_{i,i} > 0$, then x_i is an objective that is minimized; if $w_{i,i} < 0$, then x_i is an objective that is maximized. The magnitude of $w_{i,i}$ indicates the objective weight. It is assumed that the designer has properly scaled his or her design objects so that the choice of units for any particular design object does not alter the results of the optimization. The conditions that determine a design object's behavior are summarized in Table 5.1.

Importantly, *P1* and *P2* will yield the same Pareto frontiers; however, by describing the multiobjective optimization problem with *P2*, formulations are more easily manipulated, allowing

Table 5.2: Required changes in terms of lines of pseudocode to add, delete, or mutate design objects in *P1* and *P2*. Numerical value changes are not counted.

Action	**Design Object**	***P1***	***P2***
Add Design Object (New)	Objective (Independent)	5	5
	Objective (Dependent)	3	6
	Constraint	4	5
	Variable	4	4
	Parameter	2	4
	Sub Total	18	24
Delete Design Object	Objective	1	0
	Constraint	1	0
	Variable	4	4
	Parameter	2	4
	Sub Total	8	8
Modify or Mutate Design Object (Existing)	Constraint to Objective	1	0
	Variable to Objective	1	0
	Parameter to Objective	7	0
	Parameter to Variable	6	0
	Inequality Constraint to Equality	3	0
	Add Bound to Constraint	2	0
	Sub Total	20	0
	TOTAL	46	32

the designer to quickly and efficiently explore all feasible and pragmatic design spaces. If, for example, the designer wants to switch a design variable to a minimized objective, then he or she simply changes the corresponding value in the $\mathbf{w}$ matrix—no additional programming is necessary. Likewise, if he or she wants to change a parameter to a design variable, only the values in $\mathbf{y}_l$ and $\mathbf{y}_u$ need to be changed. To illustrate, consider Table 5.2, which presents the number of required line changes to generic pseudocode (see Appendix A) to modify an optimization problem formulated with *P1* and *P2*. The number of line changes are tallied for adding and deleting design objects from the formulation, as well as modifying or mutating existing design objects. Changes to values in the code were not counted, as such actions are considered trivial; for example, if the upper bound on a design variable changes from a value of 3.50 to 4.00, this is not included in the tally. As shown, on average, *P2* requires 14 less lines of code to change than *P1*.

There are some drawbacks to the dynamic optimization formulation. The required number of changes to lines of pseudocode for adding a design object is greater for *P2* than for *P1*. However, this discrepancy will be often be compensated for by the decreased number of changes to lines of pseudocode for modifying that new object after it has been created. Because a designer will often have little prior experience with this new design object in his or her formulation, modifications are expected often, mitigating to some extent the negative aspect of this drawback. Also, depending on the optimization algorithm used, computational efficiency may decrease. Since *P2* minimizes over all independent design objects ($\mathbf{y}$) including those that act as fixed design parameters, many gradient-based optimization algorithms will attempt to perturb fixed independent design objects and waste function calls. While any computational inefficiency is obviously undesirable, this is generally not a significant problem during conceptual design, because the models are typically computationally inexpensive (see Chapter 2). Moreover, other non gradient-based algorithms such as simulated annealing or genetic algorithms will see no significant efficiency losses. A closer look at the benefits and drawbacks of *P2* is provided in Chapter 8.

CHAPTER 6. EXPLORATION METRICS

To aid the designer during the exploration process, several exploration metrics are provided in this chapter. These metrics help the designer to determine (i) how well the exploration process has expanded the formulation space, (ii) how much improvement to objectives has been added through formulation space exploration, and (iii) when the exploration process is no longer diverging. These metrics loosely correspond to metrics of ideation effectiveness, as proposed by Shah and Vargas-Hernandez [76]—namely novelty, variety, quality, and quantity. The difference here is that design alternatives are being explored rather than new design concepts being generated. These metrics describe improvements made to the formulation space in terms of novelty, preferred variety, and quality.

For each of the metrics introduced in this section, there is an assumed baseline formulation objective space, denoted with the superscript $[\ ]^{(0)}$. This benchmark design space is the first to comprise the formulation space. If there is no improvement over the benchmark, then the value of the metric is zero. Higher values of each metric indicate improvement. Feasibility is assumed in formulation space exploration; by definition, a space that is not pragmatic or valid is not included in the formulation space.

6.1 Novelty

Novelty, as defined by Shah and Vargas-Hernandez [76], is a measure of how well the exploration process expands the search into regions that are *not perceived to be within the design space*. Thus, during formulation space exploration any region of the formulation space outside the original design objective space is considered novel. The metric for formulation space novelty, $\underline{M}_n$, is given by

$$\underline{M}_n = \frac{\|\underline{\Omega}\| - \|\Omega^{(0)}\|}{\|\Omega^{(0)}\|} \tag{6.1}$$

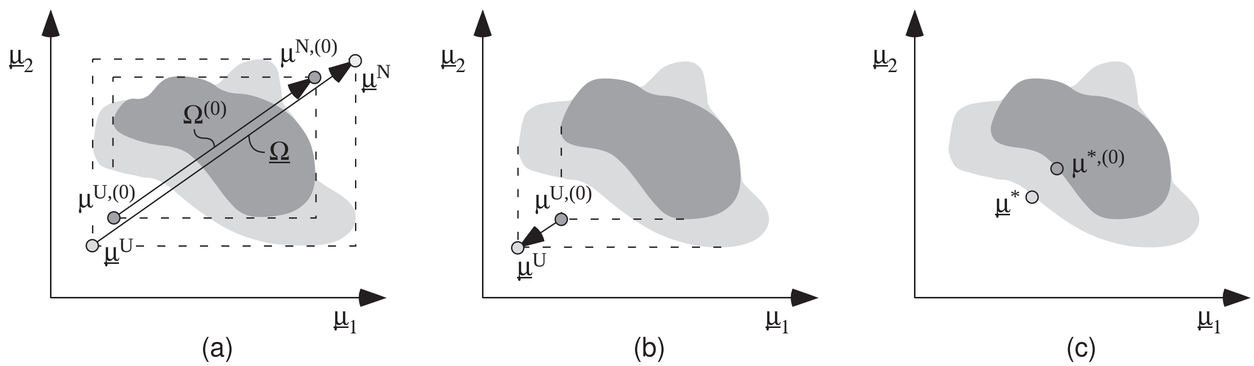


Figure 6.1: (a) Depiction of the novelty metric. (b) Depiction of the preferred variety metric. (c) Depiction of the quality metric. The darkly shaded regions are the original design space, and the lightly shaded regions represent the formulation space.

where $\underline{\Omega}$ is the diagonal of the hypercube containing the entire formulation objective space, $\Omega^{(0)}$ is the diagonal of the hypercube containing the original design objective space, and in general $\Omega^{(i)}$ is the diagonal of the hypercube containing the *i*-th space [74], or

$$\Omega^{(i)} = \mu^{N,(i)} - \mu^{U,(i)} \tag{6.2}$$

and $\mu^{N,(i)}$ and $\mu^{U,(i)}$ are the nadir point and utopia point, respectively, for the *i*-th space. This is depicted graphically for a two-dimensional formulation space in Fig. 6.1(a). The darker shaded region represents the original design space within the formulation space, shown as the lighter shaded region. The hypercubes containing both the original design objective space and the formulation space are also shown—M_n measures the difference in the vector lengths that connects the utopia and nadir points of both spaces. If design preferences are truly known, $\underline{M}_n$ is not highly valued, as it can reward exploration into regions that are not desirable. If the designer has an interest in divergently exploring a product's design space, however, this metric provides him or her with a way to quantify to what extent new design alternatives are being discovered due to the exploration process. Even if a new optimal solution is not found, the designer may gain confidence in his or her original solution after having seen the potential benefits and drawbacks of other, previously unconsidered, designs.

6.2 Preferred Variety

Preferred variety is a measure of how well the formulation space expands into desirable regions. The metric for preferred variety, $\underline{M}_v$, is given by

$$\underline{M}_v = \frac{\|\underline{\mu}^U - \mu^{U,(0)}\|}{\|\mu^{U,(0)}\|} \tag{6.3}$$

Preferred variety is depicted graphically for a two-dimensional space in Fig. 6.1(b). In the figure, as the vector between $\underline{\mu}^U$ and $\mu^{U,(0)}$ grows in length, $\underline{M}_v$ also increases. If objectives are constrained to be positive, $\underline{M}_v$ will be bounded between 0 and 1; otherwise, values greater than 1 for $\underline{M}_v$ will be possible, indicating the percent improvement of the formulation space over the initial space. In situations where a designer has not yet settled on specific objective weights, this metric may be of particular interest to him or her as a means of showing overall improvement of the Pareto frontier.

6.3 Quality – Best Design Alternative

The metric for the quality of the exploration process, $\underline{M}_q$ is measured in terms of the "best" design alternative (see Fig. 6.1(c)), or

$$\underline{M}_q = \frac{J(\mu^{*,(0)}) - J(\underline{\mu}^*)}{|J(\mu^{*,(0)})|} \tag{6.4}$$

where $\mu^{*,(0)}$ is the best design alternative at the beginning of the exploration process, $\underline{\mu}^*$ is the best design alternative at the end of formulation exploration, and J is an aggregate objective function (AOF). Many methods exist for formulating the AOF such as weighted sum methods [72], compromise programming methods [41], and physical programming [71]. The most suitable method for each specific problem is determined by the designer. If an AOF is never formally defined, then this metric can be calculated using an even weight for all objectives, essentially assigning values to solutions based upon their Euclidean distances to the origin.

6.4 Exploration Value to Effort Ratio

Formulation space exploration must be advantageous in order to be useful. As the name suggests, the exploration value to effort ratio is defined as

$$\varepsilon \equiv \frac{\text{value}}{\text{effort}} \tag{6.5}$$

where value can be defined as any individual metric or combination of the metrics defined above. For instance, value could be assessed by the best design alternative with Eq. 6.4. The effort can be measured in terms of coding complexity [77,78], number of function calls in the optimization algorithm, or in computation time. Regardless of method, there is an associated cost with formulation space exploration; therefore, it is pivotal to monitor the metrics described in this section to ensure that value is being added through the exploration process. When the exploration process ceases to produce novel design alternatives, improve objective values, or discover new "best" designs according to the user-defined AOF, the formulation space is no longer diverging and subsequent exploration will likely decrease ε.

A noted weakness in the provided metrics is their dependency on the original formulation. Each metric is scaled to show an improvement with respect to that formulation, which means that identical final formulation spaces may exhibit differing values for these metrics, due to differences in the original formulations from which they sprang. This dependency suggests that these metrics are more effectively used as a means of recognizing the amount of improvement of a particular formulation space, rather than as a comparison between multiple exploration processes.

The metrics were created to be a design tool—the designer can review these metrics after each optimization formulation to determine if "value" is being added by the exploration process. For example, this can be done on a run-sequence chart; when the metrics flatline in the chart, the designer will know that the formulation space may no longer be diverging in a useful manner. Since the designer is reviewing the results of the optimization process at each iteration, the designer literally becomes part of the optimization loop, using his or her judgement to help guide the exploration process. More information on how to use these metrics as a design tool is provided in Chapter 8.

CHAPTER 7. USE SCENARIOS FOR FORMULATION SPACE EXPLORATION

In this chapter, three scenarios are presented for using the dynamic optimization problem formulation presented in Chapter 5. All three scenarios can be encountered during conceptual design, after at least one design concept has been developed. The applicability of each scenario is governed by the amount of information a designer truly knows at that point in the design process. Sec. 7.1, discusses a scenario where the designer has solidified the objectives of the project and is ready for concept selection. Sec. 7.2 describes a situation where the designer knows little about the design objectives and is more interested in the formulation space as a whole rather than any particular Pareto frontier. And Sec. 7.3 investigates how to explore regions of infeasibility, with the intent of learning more about the design, its trade-offs, and potential future design possibilities.

7.1 Scenario 1: s-Pareto Generation for Multiple Formulations

Perhaps the most obvious use for an optimization problem is to converge to an optimal solution. In this section, the dynamic multiobjective optimization approach is combined with an s-Pareto generation and selection strategy presented by Mattson and Messac [4]. Consider the two formulation spaces shown in Fig. 7.1. An s-Pareto frontier is defined as the Pareto optimal

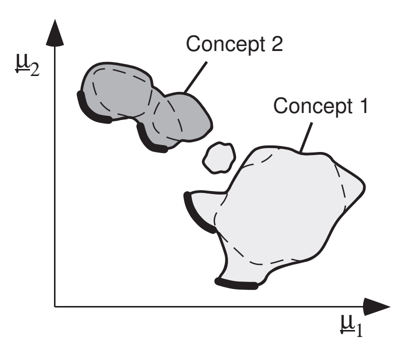


Figure 7.1: Design objective formulation spaces for two concepts are shown. The resulting s-Pareto frontier is outlined in bold and spans both concepts and multiple formulations.

solutions for a set of concepts. In this case, however, an s-Pareto frontier in a formulation space is shown, because it is the Pareto optimal solutions for the set of all formulations and concepts. This is evident in the figure because the bolded line, representing the s-Pareto frontier, spans multiple concepts and formulations. With the s-Pareto frontier defined, concept selection can proceed with a qualitative or quantitative analysis of concept goodness as described in [79].

The optimization problem given by *P2* (Eqs. 5.1–5.6) needs to be modified to account for multiple concepts. The generic, dynamic multiobjective optimization problem capable of comparing multiple concepts is given by Problem 3 (*P3*):

$$\min_{k} \left\{ \min_{\mathbf{y}^{(k)}} \left\{ \mu_1^{(k)}(\mathbf{x}^{(k)}),\ \mu_2^{(k)}(\mathbf{x}^{(k)}),\ \ldots,\ \mu_{n_x^{(k)}}(\mathbf{x}^{(k)}) \right\} \right\} \quad (n_x^{(k)} \geq 2) \tag{7.1}$$

where the superscript $[\ ]^{(k)}$ indicates that $[\ ]$ is associated with formulation or concept k. Equations 5.2–5.6 from *P2* are still valid here for *P3*, although each equation will be specific to the formulation or concept k. Note that Eq. 7.1 is a minimization problem nested within another minimization problem; in other words, solving *P3* will find the minimum of all formulation or concept minima. By using *P3*, the designer is able to diverge the design space by considering the formulation spaces of all concepts, and then converge or select the concept that is most likely to meet the final design specifications.

The objectives minimized in *P3* are *set objectives*, meaning that they are comparable across all formulations and concepts. The inputs to the concept models that generate set objectives may be unique. For example, the required variable inputs needed to calculate the mass of a bevel gear and a spur gear may be different, yet the mass of the two types of gears is comparable. Additionally, it is possible for a formulation or concept to have one or more objectives that are specific to the formulation or concept. To illustrate, consider a design concept that contains a hazardous material. It may be necessary to maximize the safety of this concept, whereas other design concepts generated may not contain the hazardous material, obviating the need to maximize safety. These formulation and concept specific objectives are easily included as constraints in $\mathbf{x}$. More information on how to handle formulation or concept specific objectives can be found in [4].

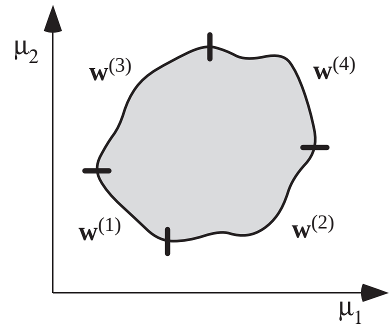


Figure 7.2: The results of four different optimization formulations are overlaid on the design objective space.

7.2 Scenario 2: Formulation Space Boundary Exploration

A second scenario where the dynamic formulation allows the engineer to explore and learn more about a product's design space is through formulation space *boundary* exploration. If design objectives and preferences are truly unknown, which is often the case in early design, then finding an s-Pareto frontier for a set of concepts is less meaningful than finding the full boundaries of the feasible space for a given formulation. For example, when designing a vehicle for use in different environments, it may be necessary to either minimize trunk space or maximize trunk space, depending on the final function of the vehicle – either extreme could be considered ideal. In this bimodal scenario, information about the design space as a whole may be better suited to help guide the designer toward a final decision about the true objectives of this design.

Formulation space boundary exploration is fully possible with either *P1* or *P2*, however, this procedure will only be explicitly presented with *P2*. Recall that in *P2*, objectives are controlled by the values along the diagonal in **w**. For a two-dimensional problem of a single optimization formulation, such as the one seen in Fig. 7.2, it is possible to obtain the boundary of the design objective space using four different **w** matrices in *P2* and the normal boundary intersection method or a modified normal constraint method [80,81]. For example, in $\mathbf{w}^{(1)}$ in the figure, both objectives are minimized (i.e., $\mathbf{w}_{i,i}$ corresponding to μ_i for both objectives is 1); this results in an optimization problem that produces the lower left boundary of the design space. The three remaining boundaries can be obtained by toggling the requisite values in **w** between 1 and -1. A similar process can be used to find the boundaries of the formulation space.

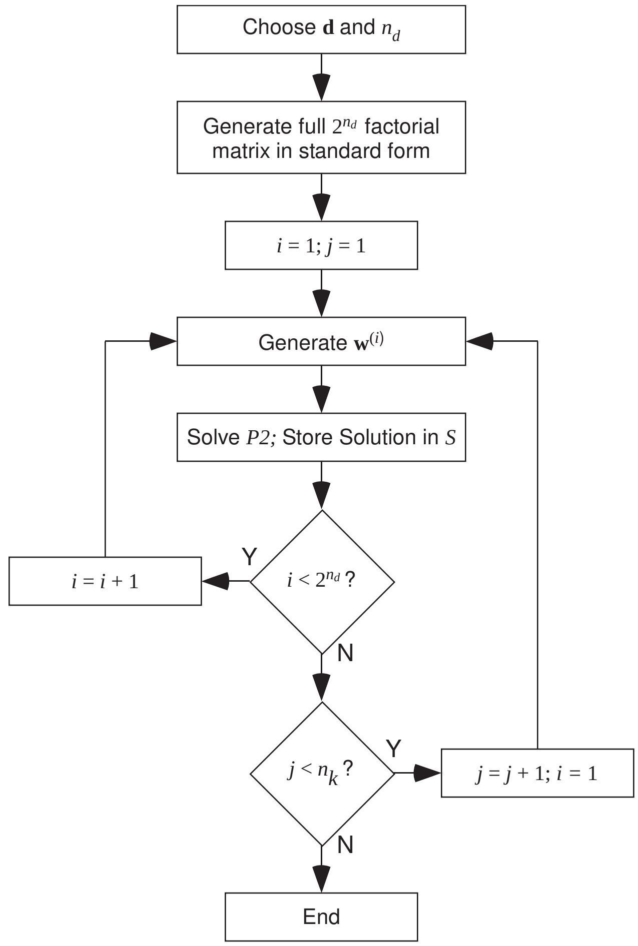


Figure 7.3: A flow chart is shown for determining the boundaries of a formulation space using the dynamic optimization formulation.

The general process for finding the boundaries of the formulation space is shown in Fig. 7.3. First, the designer chooses the design objects of interest from the vector **x**, the total number of which is n_d, and stores the indices that correspond to **x** in a vector, **d**. Second, the designer generates a 2^{n_d} two-level, full factorial matrix, **f**, in standard form. For example, if $n_d = 2$, the following matrix would be displayed.

$$\mathbf{f} = \begin{bmatrix} -1 & -1 \\ 1 & -1 \\ -1 & 1 \\ 1 & 1 \end{bmatrix} \tag{7.2}$$

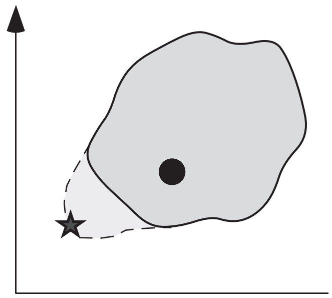


Figure 7.4: An original design objective space is darkly shaded, with a feasible point shown as a circle on the interior of this space. A star represents a design point of interest that is not feasible. The lighter shaded, dashed region represents a new space obtained through inverse optimization.

Next, *P2* is executed 2^{n_d} times in a loop. For every iteration in the loop, $\mathbf{w}^{(i)}$ is generated by setting $w_{l,l}$ to the the i-th row and the m-th column of $\mathbf{f}$, where l is the m-th entry in $\mathbf{d}$. For every loop, the solution is added to the set S and i is incremented. Once i is greater than 2^{n_d}, the process is repeated for any remaining concepts or formulations, the total number of which is n_k. This allows the designer to see the extreme boundaries of the formulation space with respect to any combination of objectives.

7.3 Scenario 3: Inverse Optimization

A third scenario for using the dynamic optimization problem formulation is to perform design feasibility studies, or inverse optimization. Consider the feasible design space shown as the dark shaded region in Fig. 7.4. If the designer wants to see designs near the black circle, he or she could easily run an optimization (using *P1* or *P2*) that minimizes the Euclidean distance to that point in the design objective space [52]. However, if the designer would like to see designs near the black star in Fig. 7.4, which is located outside of the feasible design space, then the problem constraints need to change ($\mathbf{y}_l$, $\mathbf{y}_u$, $\mathbf{z}_l$ and $\mathbf{z}_u$ need to be modified) to extend the searchable space towards the star. This scenario could occur if a designer highly desires a particular performance in the product, and is willing and able to compromise some of the constraints of the design. For example, consider a project where a design team has been given a certain budget, which they understand to be a constraint. The team wants to know, however, how much more it would cost to get to a particular performance level that they currently cannot reach, given the monetary constraints of the budget. If the increase in cost is fairly small for a significant increase in performance, this may justify a request for a change in budget, or a change in the constraints that they have been given. In this manner, the *optimization formulation itself becomes a part of the optimization*.

Presented here is an optimization formulation with two objectives functions to explore infeasible regions of interest as Problem 4 (*P4*):

$$\min_{\mathbf{x}_l,\mathbf{x}_u} \left\{ f_1(\mathbf{x}_l,\mathbf{x}_u,\mathbf{v}),\ f_2(\mathbf{x}_l,\mathbf{x}_u,\mathbf{x}_l^{(0)},\mathbf{x}_u^{(0)}) \right\} \tag{7.3}$$

subject to the side constraints

$$x_{l,i}^{(-)} \leq x_{l,i} \leq x_{l,i}^{(+)} \quad \{i = 1,...,n_x\} \tag{7.4}$$

$$x_{u,i}^{(-)} \leq x_{u,i} \leq x_{u,i}^{(+)} \quad \{i = 1,...,n_x\} \tag{7.5}$$

and

$$f_1(\mathbf{x}_l,\mathbf{x}_u,\mathbf{p}) = \min_{\mathbf{y}} \|\gamma(\mathbf{y}) - \mathbf{v}\| \tag{7.6}$$

$$f_2(\mathbf{x}_l,\mathbf{x}_u,\mathbf{x}_l^{(0)},\mathbf{x}_u^{(0)}) = \|(\mathbf{x}_l - \mathbf{x}_l^{(0)})\| + \|(\mathbf{x}_u - \mathbf{x}_u^{(0)})\| \tag{7.7}$$

where the superscripts $[\,]^{(-)}$ and $[\,]^{(+)}$ indicate a lower or upper bound on [ ], respectively, and the superscript $[\,]^{(0)}$ indicates that [ ] is from the original formulation (i.e., the formulation that defines the feasible objective space – the dark shaded region in Fig. 7.4). The vector $\mathbf{x}_l$ is the concatenation of $\mathbf{y}_l$ and $\mathbf{z}_l$, or $\mathbf{x}_l = [\mathbf{y}_l;\mathbf{z}_l]$; likewise, $\mathbf{x}_u = [\mathbf{y}_u;\mathbf{z}_u]$. The vector $\mathbf{v}$ represents a point of interest in the current infeasible objective space, or the star in Fig. 7.4, and $\gamma(\mathbf{y})$ is the set of objectives in $\mathbf{x}$ that correspond to $\mathbf{v}$. According to Eq. 7.3, the designer attempts to minimize the Euclidean distance between $\gamma(\mathbf{y})$ and $\mathbf{v}$ while also minimizing the changes made to the original optimization formulation. As with most optimization problems, proper scaling of the design objects will produce better results; this is especially critical when calculating f_2 with Eq. 7.7, as a relatively small change in one constraint could be large in comparison to another.

Successfully solving *P4* will result in a Pareto frontier of solutions, each of which points to an optimization formulation that searches in a design objective space that includes the point of interest, if possible. In other words, this is an inverse optimization process, where the designer picks a point of interest, and the optimization formulation that can find that point is returned. Ultimately, the designer will discover the minimum cost to obtain a desired performance.

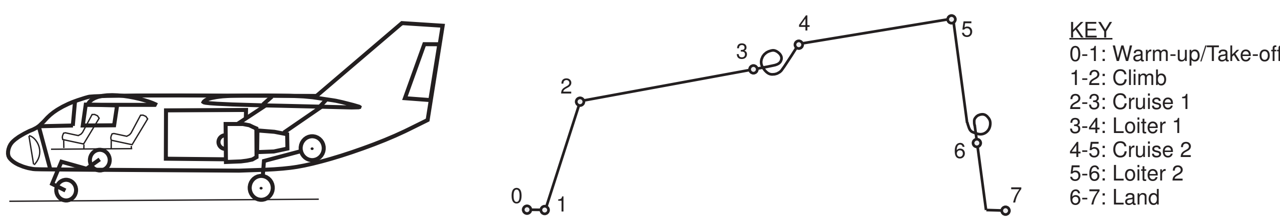
CHAPTER 8. CASE STUDIES

In this chapter, two case studies are presented. The first example in Sec. 8.1 is used to compare the performance of the standard multiobjective optimization problem (Problem 1 from Chapter 3) to the dynamic optimization problem (Problem 2 from Chapter 5). The second example in Sec. 8.2 illustrates how formulation space exploration can be used in various scenarios to produce valuable information for designers.

8.1 Case Study: Conceptual Sizing of an Aircraft

The purpose of this case study is to illustrate how to use formulation space exploration presented in Chapter 4 to search a product's design space in both a divergent and convergent manner during conceptual design. As part of the study, the benefits and limitations of the dynamic optimization problem formulation from Chapter 5 are quantified and used to compare its performance with the standard multiobjective optimization problem formulation.

The case study is based upon the conceptual sizing problem of an antisubmarine warfare aircraft, initially presented in *Aircraft Design: A Conceptual Approach* [3], where a rudimentary analytical model for sizing any aircraft from a conceptual sketch is developed from statistical and historical data. The conceptual sketch for this case study is shown in Fig. 8.1(a), and the mission



(a) (b)

Figure 8.1: (a) Antisubmarine warfare aircraft conceptual sketch. (b) Flight mission details. Images are adapted from *Aircraft Design: A Conceptual Approach* [3]

Table 8.1: Summary of inputs and outputs to the antisubmarine warfare aircraft concept model.

Inputs	**Outputs**
Weight of Payload	Wetted Aspect Ratio
Weight of Crew	Maximum Lift to Drag Ratio
Cruise Speed	Lift to Drag Ratio During Cruise
Wing Aspect Ratio	Lift to Drag Ratio During Loiter
Wetted Area Ratio	Unknown Fuel-Weight Fractions of Mission
Cruise Range	Take-off Weight
First Loiter Time	
Second Loiter Time	
Known Fuel-Weight Fractions of Mission	
Empty Weight Fraction Model Coefficients	

profile in (b). The model inputs and outputs are summarized in Table 8.4. A total of 17 independent design objects and 11 dependent design objects are included in the model. This model is well suited for formulation space exploration because it is a well-known example of conceptual design, where the expressed intent for its use is in *evaluation and refinement, with the customer, of the design requirements* [3].

Eight optimization problem formulations are created here using *P2* as the template; each formulation is summarized below:

- **Formulation 0** - Maximize cruise range and minimize take-off weight subject to lower and upper constraints on all model outputs and by allowing the wing aspect ratio and cruise range to vary between side constraints. All other model inputs are fixed independent design objects (i.e., fixed design parameters).
- **Formulation 1** - Same as Formulation 0, except the weight of the payload varies.
- **Formulation 2** - Same as Formulation 1, except the wetted area ratio varies.
- **Formulation 3** - Same as Formulation 2, except the weight of the payload is added as a maximized objective and the total fuel-weight fraction is added as a minimized objective.
- **Formulation 4** - Same as Formulation 3, except the total fuel-weight fraction is removed as a minimized objective.

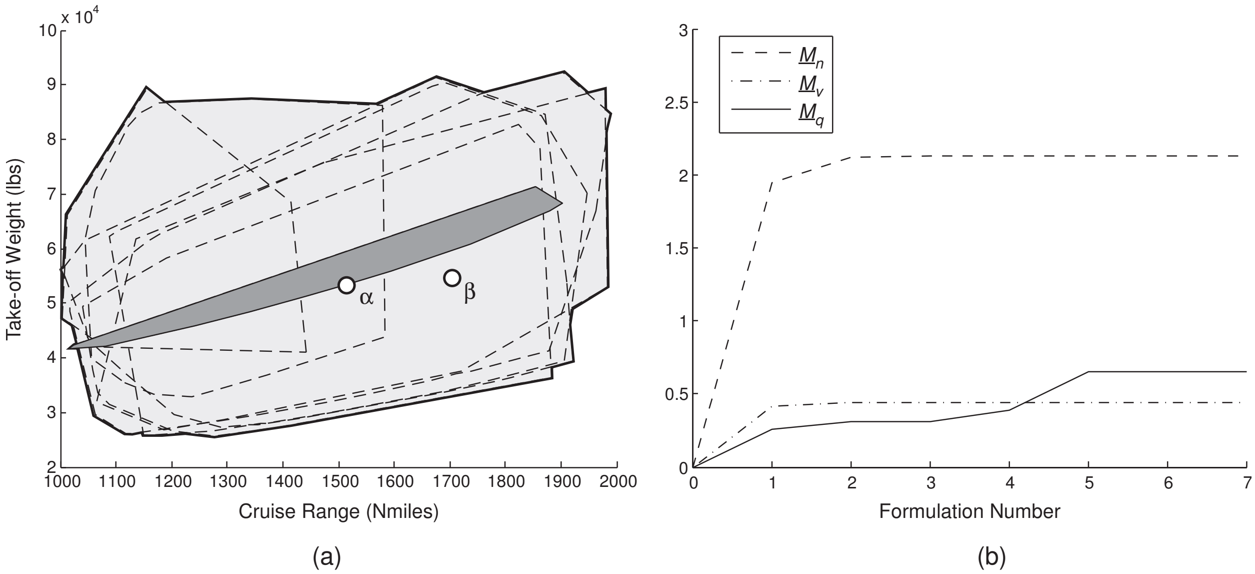


Figure 8.2: (a) Formulation objective space for cruise range and take-off weight. Darker shaded region is baseline formulation. (b) Novelty, preferred variety, and quality metrics versus formulation number.

- **Formulation 5** - Same as Formulation 4, except the input values are changed to represent a composite aircraft instead of one fabricated out of aluminum.
- **Formulation 6** - Same as Formulation 4, except the cruise speed is allowed to vary and change input values to simulate a low-bypass turbofan engine rather than a high-bypass turbofan.
- **Formulation 7** - Same as Formulation 4, except the first mission loiter time is allowed to vary.

The results from the formulation exploration process are shown in Fig. 8.2. In (a), a two-dimensional slice of the formulation space is shown for cruise range and take-off weight (shown as the lighter shaded region, with the baseline formulation shown as the darker shaded region and all other formulations shown as dashed lines); in reality, the formulation space is three-dimensional since there are three objectives in this problem. The exploration metrics from Chapter 6 have been plotted in (b). The novelty metric at the end of the exploration process is 2.13, meaning that the diagonal of the hypercube containing the entire formulation space is 2.13 times larger than the baseline. The preferred variety metric ends at a value of 0.44, meaning that the distance between the formulation utopia point and the baseline utopia point 0.44 times the distance of the baseline utopia point. The quality metric is at 0.65, indicating that the AOF value, which in this case is determined with a weighted sum, of the formulation space is 0.65 times better than the baseline AOF value.

An important role of the exploration metrics is to help indicate when the formulation space is no longer diverging. Figure 8.2(b) shows that the slope of the exploration metrics after Formulation 5 is zero; this likely indicates that no new solutions are being found by the exploration process (i.e., the formulation space is no longer diverging). Note that at Formulation 5, the quality metric ($\underline{M}_q$) reached its maximum value. If Formulation 5 had been formulated first in the exploration process, there would have been no improvement in $\underline{M}_q$ in subsequent formulations; in other words, the slope of $\underline{M}_q$ would never increase and the value of $\underline{M}_q$ would remain at zero. However, there may have been improvement in the other metrics. If the designer was only interested in $\underline{M}_q$, then they may not have tried the remaining formulations.

The benefits from formulation space exploration are not only measured by the metrics above, but by what information is gleaned from the process. It is clear that various trade-off studies are possible through formulation space exploration. Obtaining the Pareto frontier for the formulation space provides a rich set of design alternatives from which the designer can "shop" for the most suitable solution [51]. For example, consider points α and β in Fig. 8.2(a). Point α is a design alternative from the first optimization formulation, and point β is a design alternative from the three-dimensional formulation space Pareto frontier, originating from Formulation 5 where a composite aircraft is simulated. These designs are juxtaposed in Table 8.2. While the weight of the payload and take-off weights are similar, the range for Design β is significantly greater.

Formulation space exploration can be done with *P1* or *P2*. However, with *P2* it is simpler to formulate and reformulate the optimization problem. According to Table 5.2, the formulation space exploration process presented in this section would require 27 lines of code to be changed if *P1* is used, whereas with *P2* no lines of code need to be changed (only the values of $\mathbf{y}_l$, $\mathbf{y}_u$, $\mathbf{z}_l$, $\mathbf{z}_u$, and $\mathbf{w}$ need to be changed). However, as stated in Chapter 5, there are some limitations. To illustrate, the baseline problem (Formulation 0) is formulated using *P1* and *P2* and is executed with various optimization algorithms including two gradient based methods with a weighted-sums AOF: sequential quadratic programming (SQP) and Interior-point; and an evolutionary algorithm:

Table 8.2: Details about two design alternatives in the formulation space.

	Weight of Payload (lbs)	Wing Aspect Ratio	Wetted Area Ratio	Range (Nm)	Weight Fraction Coefficient	Wetted Aspect Ratio	Max Lift to Drag Ratio	Total Fuel Weight Fraction	Takeoff Weight (lbs)
Design α	10,000	8.40	5.50	1520	1.00	1.53	16.96	0.36	53,639
Design β	10,154	8.17	5.19	1715	0.95	1.57	17.24	0.39	54,184

Table 8.3: Comparison of computation performance for *P1* and *P2* on an Intel Core 2 Quad 2.67GHz processor.

	SQP		Interior-Point		Genetic Algorithm	
	Time (s)	Function Count	Time (s)	Function Count	Time (s)	Function Count
P1	2.37	48	2.66	77	28.60	1240
P2	9.08	288	11.18	369	28.07	1240

genetic algorithm. The computation time and function call count for each algorithm are summarized in Table 8.3. As shown, *P1* outperforms *P2* when using the gradient-based algorithms; however, no significant difference is seen in the evolutionary algorithm. The baseline problem includes 2 design variables as part of 17 independent design objects. When the ratio of design variables to independent objects is low, as it is here, *P2* does not perform as well as *P1* with gradient-based algorithms. As the ratio approaches one, the computational efficiency differences between *P1* and *P2* diminishes. For this example, the computation time difference between *P1* and *P2* is only a few seconds.

Another way to compare the performance between *P1* and *P2* is with the exploration value to effort ratio (see Chapter 6). In this example, the value in Eq. 6.5 is the sum of all exploration metrics above—$\underline{M}_n$, $\underline{M}_v$, and $\underline{M}_q$—multiplied by 100 for proper scaling. The value added is the same whether *P1* or *P2* is used. Effort is approximated as the time it takes to code each formulation plus the computation time. This method for approximating effort does not take into account the time spent interpreting results, planning future formulations, etc. Nonetheless, it is sufficient for the purpose at hand. Assume that it takes 10 seconds to change each line of code (27 lines for *P1* and 0 for *P2*). Using the genetic algorithm computation times from Table 8.3, the exploration value to effort ratio for *P1* is 1.08, and for *P2* is 11.47. If instead an SQP algorithm is used in conjunction with the normal constraint method [74] to generate 30 Pareto optimal solutions, the value to effort ratio for *P1* is 0.94, and for *P2* is 1.18. In each case, *P2* outperforms *P1*.

8.2 Case Study: Impact Driver Design

The purpose of this case study is to illustrate how to use the dynamic optimization exploration processes under the different scenarios presented in Chapter 7. Although this case study is anecdotal in nature, it illustrates several important points: (i) By searching the formulation space, designers are able to search the design space in both a divergent and convergent manner. (ii) Formulation space exploration requires the human designer to be intimately involved in the search process, allowing their judgement and rational decision-making capabilities to guide the search. (iii) Using the dynamic optimization problem formulation promotes design exploration. The focus here is not to defend the practicality of the proposed product, but rather to show how the methods presented in this thesis could be used in the development of a new product.

The case study is based upon a proposed new type of impact driver, which is a specialized tool that applies high torque to fasteners by the means of a hammer mechanism. The novel aspect of this new type of impact driver is depicted in Fig. 8.3. On the left side of the figure, a backpack holding several batteries is shown. The batteries connect to a power cord which runs from the backpack, down the user's arm and into a special glove with electrical contacts embedded in the palm of the glove (shown on the right in the figure). There are corresponding electrical contacts on the impact driver. Thus, a complete, electrical circuit is made when the user grabs the impact driver with the glove on.

The goals of the design are to (i) reduce arm fatigue for those who use the impact driver for long periods of time, such as outdoor deck fabricators, sheetrock hangers, or general construction workers; (ii) increase the battery life between charges (more batteries can fit in a backpack than directly on a typical impact driver); and (iii) maintain the mobility of a cordless impact driver. For the remainder of the case study, attention will be directed towards how a designer might develop an impact driver to accompany the backpack and glove – specifically, how to design a DC motor

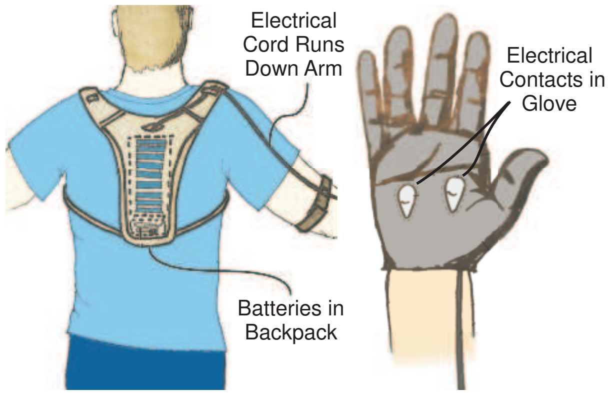


Figure 8.3: General idea for new type of impact driver. A backpack holds several batteries, which connect to a special glove via a power cord. The glove has electrical contacts that correspond and connect power to an impact driver. Image courtesy of Garrett Bodily at Brigham Young University.

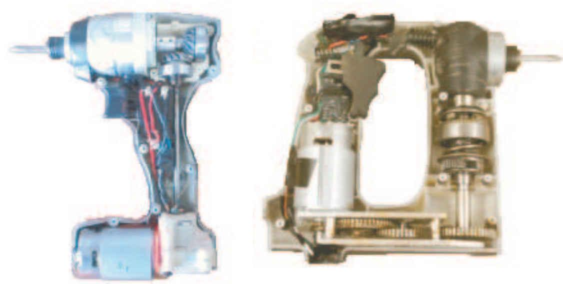


Figure 8.4: Two existing functional prototypes for new impact driver – the left prototype corresponds to Concept 1 in the case study and the right prototype is represented by Concept 4.

impact driver with *no* battery attachment. Two different groups of engineering graduate students at Brigham Young University (BYU) designed and built functional prototypes of this novel type of impact driver. Both prototypes are shown in Fig. 8.4; the results of the exploration process will be compared with these designs.

Five impact driver concepts have been generated that could potentially fulfill the design specifications of this case study, shown as Concepts 1 through 5 in Fig. 8.5. In each case, the geometry and product architecture is altered, and the drive train components are added or subtracted to achieve the desired goals. Each concept uses the same impact assembly, which has been designed previously and is the existing impact assembly for a 12V Hitachi (model WH10DFL) impact driver. Each concept is described in greater detail below:

- **Concept 1** - In this concept, the impact assembly (I) is oriented horizontally above the trigger assembly (T). Two sets of bevel gears (G) connect the impact driver to the motor (M), which is oriented horizontally and located where batteries are typically found on most commercially available impact drivers.

- **Concept 2** - This concept is similar to Concept 1; however, the motor is oriented vertically. Only one set of bevel gears is needed to connect the motor to the impact assembly. Additionally, a counterweight (W) is added to the design.

- **Concept 3** - In this concept, the motor is directly attached to the impact assembly, obviating the need for any bevel gears. The counterweight from Concept 2 is included to help improve balance.

- **Concept 4** - In this concept, the impact assembly is oriented vertically, with a set of bevel gears at the output to allow the user to drive fasteners horizontally. A gear train consisting of four spur gears (S) connects the impact driver to the motor, which is also oriented vertically. The trigger assembly is located directly above the motor.

- **Concept 5** - This concept is similar to Concept 4; however, the motor is located directly below the trigger assembly and the impact assembly. No gear train is needed in this concept as the motor is directly inline with the impact assembly.

Five separate models have been developed to analyze the concepts. The model inputs ($\mathbf{y}$) and outputs ($\mathbf{z}$) are summarized in Tab. 8.4. For every model, the origin is defined as the upper corner of the trigger assembly that faces the front of the impact driver (see Fig. 8.5); assume the ideal center of mass of the impact driver is at that point, which is approximately true for the commercially available 12V Hitachi impact driver. While the inputs to every model vary, each model includes a total mass estimate, a total cost estimate, torque output, speed output, as well as various other outputs of interest. With models defined for each concept, formulation space exploration can begin.

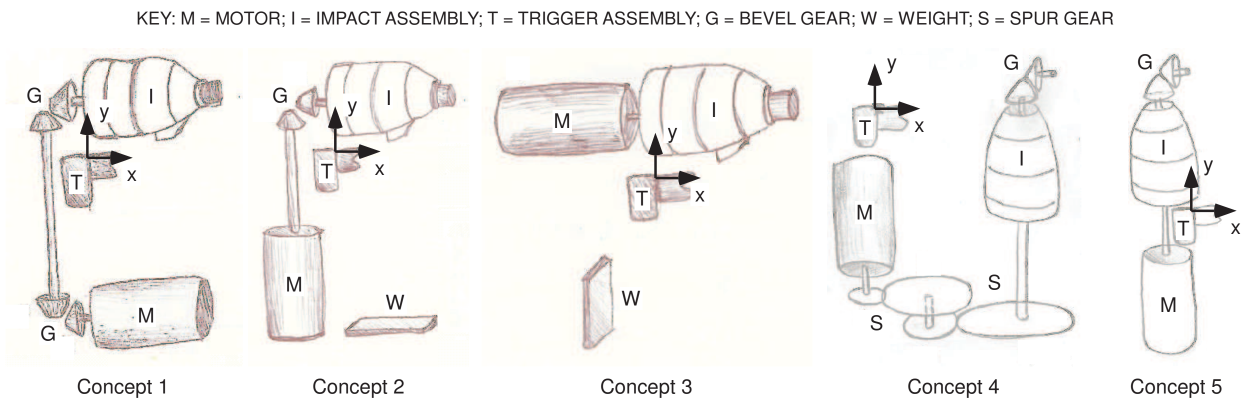


Figure 8.5: Five concepts for an impact driver with no batteries.

Table 8.4: Model inputs and outputs for the five impact driver concepts.

Concepts	**Model Input** (**y**)
1, 2, 3, 4, 5	Drive Shaft Materials
1, 2, 3, 4, 5	Drive Shaft Sizes
1, 2, 4, 5	Gear Material
1, 2, 4, 5	Gear Type
1, 2, 3, 4, 5	Motor Type
1, 2, 3, 4, 5	Motor Location
1, 2, 3, 4, 5	Impact Assembly Location
2, 3	Counterweight Location
2, 3	Counterweight Material
2, 3	Counterweight Size

Concepts	**Model Output** (**z**)
1, 2, 3, 4, 5	Drive Shaft Locations
1, 2, 3, 4, 5	Shaft Stress Constraint
1, 2, 4, 5	Gear Locations
1, 2, 4, 5	Gear Torque Constraint
1, 2, 3, 4, 5	Total Mass
1, 2, 3, 4, 5	Center of Mass
1, 2, 3, 4, 5	Total Cost
1, 2, 3, 4, 5	Number of Collisions
1, 2, 3, 4, 5	Torque Supplied
1, 2, 3, 4, 5	Torque Difference
1, 2, 3, 4, 5	Speed Supplied
1, 2, 3, 4, 5	Speed Difference

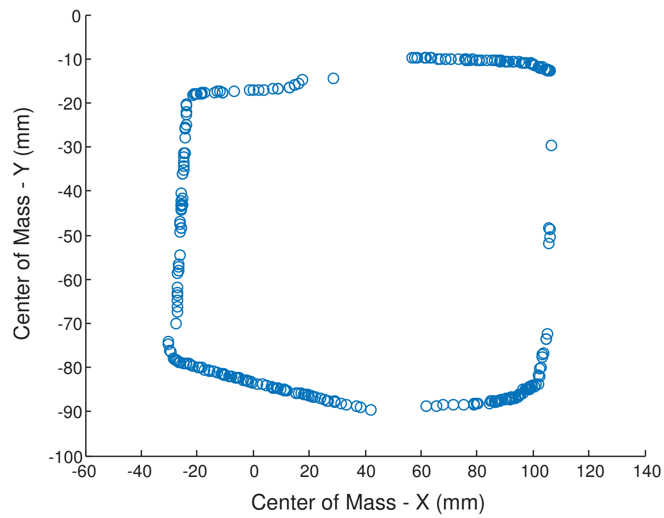


Figure 8.6: Boundary exploration for the center of mass of Concept 1.

8.2.1 Boundary Exploration

Recall that a main goal of the design is to reduce arm fatigue for those who use the impact driver for long periods of time. Thus, it is reasonable to begin formulation space exploration with the assumption that a designer would want to minimize the total mass of the impact driver and minimize the Euclidean distance between the ideal center of mass (located at the origin of the models) and the actual center of mass of each concept. Nevertheless, other objectives are still unclear at this point; it is in this scenario where boundary exploration is most useful. Concept 1 will be considered first. A preliminary optimization problem is formulated, and using the process outlined in Fig. 7.3 (with $n_d = 2$ and $n_k = 1$) boundaries for the x and y locations of the center of mass are explored. The result is shown in Fig. 8.6. The plot illustrates that the ideal center of mass (0,0) is not possible, given the current optimization formulation. Additionally, the x-location of the center of mass ranges from -30mm to 105mm and the y-location from -90mm to -10mm. Using this data, worst case scenario experiments can be run with rudimentary, physical prototypes to determine whether the x-location or y-location has a greater effect on arm fatigue and use this information in subsequent optimization formulations.

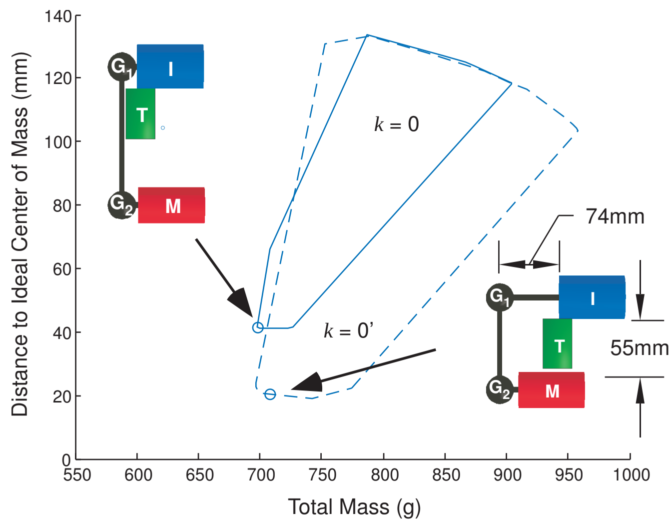


Figure 8.7: Visualization of design alternatives for two formulations.

8.2.2 Visualization

Using the results from the previous section, a new optimization problem has been formulated that now includes a third objective: minimize the x-location of the center of mass. The projection of this new 3-dimensional design objective space is plotted in a two-dimensional plane as the dashed lines in Fig. 8.7 and labeled as $k = 0'$. The x-axis in the figure is the total mass of the impact driver in the grams and the y-axis is the distance to the ideal center of mass in millimeters. An architectural layout for one design alternative on the Pareto frontier of this formulation is depicted on the right in the plot; the labels are the same as those in Fig. 8.5. Notice that the vertical drive shaft of this design is relatively distant from the trigger assembly. From a design usability standpoint, the vertical shaft should fit inside the impact driver handle, along with the trigger assembly. In this particular design alternative, it is clear that in order for both components to fit inside the handle, the handle needs to be quite large – too large, in fact, for a hand to grip easily. Moreover, the vertical distance between the impact assembly and the motor is too small. In other words, this space is not pragmatic or valid, and by definition does not contribute to the formulation space.

With what has been learned from $k = 0'$, a new optimization problem ($k = 0$) has been formulated with an added objective: minimize the distance between the trigger assembly and the vertical shaft. The constraint on the y-location of the motor has also been updated to allow more vertical space for a hand to grip the impact driver. The resulting design space is shown as the region enclosed by solid lines in Fig. 8.7. A design alternative from this Pareto frontier is depicted on the left – notice that there is no horizontal space between the vertical drive shaft and the trigger assembly, and there is adequate vertical space between the impact assembly and the motor. As seen here, visualization of optimization results is critical to effective formulation space exploration. In this case, the architectural layouts are generated by a concept analytical model, and while low in fidelity, provide adequate information to decision-makers. However, visualization of design alternatives is not always practical, nor is it possible to directly and simultaneously plot formulation spaces that exist in more than three dimensions. Optimization visualization is a topic of ongoing research, and several methods exist that could potentially facilitate formulation space exploration [52, 82–84].

8.2.3 Inverse Optimization

Suppose that a designer wants to learn the minimum amount of change to the current optimization formulation ($k = 0$) that would result in an objective space that contains the following point of interest: (mass = 600g, distance to ideal center of mass = 30mm). The star in Fig. 8.8(a) represents this point of interest. Using *P4*, an optimization algorithm is allowed to modify the current lower and upper bounds of the weight of the motor, the length of the motor, the torque output of the motor, and the shaft stresses within new ranges that are defined. These ranges are contained in $\mathbf{x}_l^{(-)}$, $\mathbf{x}_l^{(+)}$, $\mathbf{x}_u^{(-)}$, and $\mathbf{x}_u^{(+)}$. Solving *P4* results in a Pareto frontier of *optimization formulations*, shown in Fig. 8.8(b). The minimum scaled distance to the point of interest is shown on the x-axis, and the number of changes to the original optimization formulation is on the y-axis. One of these optimization formulations is highlighted as $k = 1'$. Using the lower and upper bounds on $\mathbf{x}$ from this formulation in *P2*, the design objective space is plotted in Fig. 8.8(a) with dashed lines. As shown, this space has the point of interest on its Pareto frontier, and contains most of the previous formulation ($k = 0$). Although this region is assumed to be infeasible, the minimum change to the formulation that would be required in order to obtain the objective values of the point of interest comes from lowering the upper constraint on the motor mass by 95g and the motor torque by 201 N-mm.

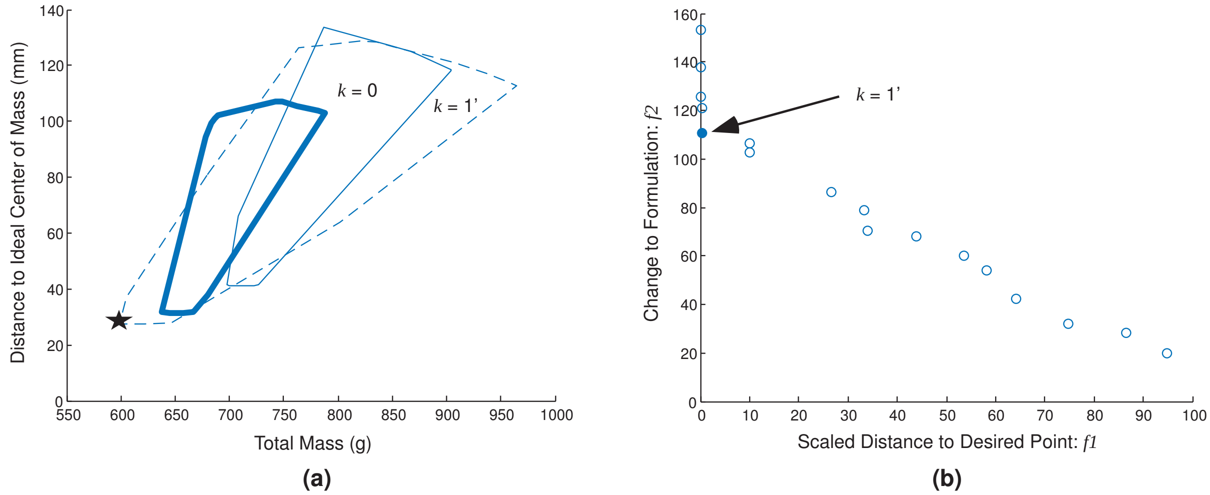


Figure 8.8: (a) A point of interest (the star) is selected, and an optimization problem is formulated to modify the constraints of the design space in thin solid lines so that it includes the point of interest. The resulting space is shown with dashed lines. Another space, shown in bold, is then formulated to approximate the results of the inverse optimization. (b) This is the Pareto frontier of the inverse optimization problem.

With this information, a different motor can be found for the design that will approximate the results of the inverse optimization. With a new motor, additional constraints in the formulation are needed to ensure that the torque and speed of the impact driver are appropriate. The design space of the new formulation ($k = 1$) is shown in bold in Fig. 8.8(a). While formulation 1 does not match the performance of formulation 1' exactly, it is noticeably better than formulation 0.

8.2.4 s-Pareto Generation and Concept Selection

Six more formulations are created for Concept 1 and shown in Fig. 8.9. As long as the designer finds each explored region to be pragmatic and useful, the union of these regions becomes the formulation space. Using the metrics developed in Chapter 6, the goodness of this formulation space exploration process can be quantified in terms of three aspects: novelty, preferred variety, and quality. Novelty is a measure of how expansive the search has been. Preferred variety indicates how well the search has expanded in useful directions. And quality indicates improvement in the "best" design as determined by an aggregate objective function. For the formulation exploration of Concept 1, the values of 0.35, 0.11, and 0.08 are achieved for novelty, preferred variety, and

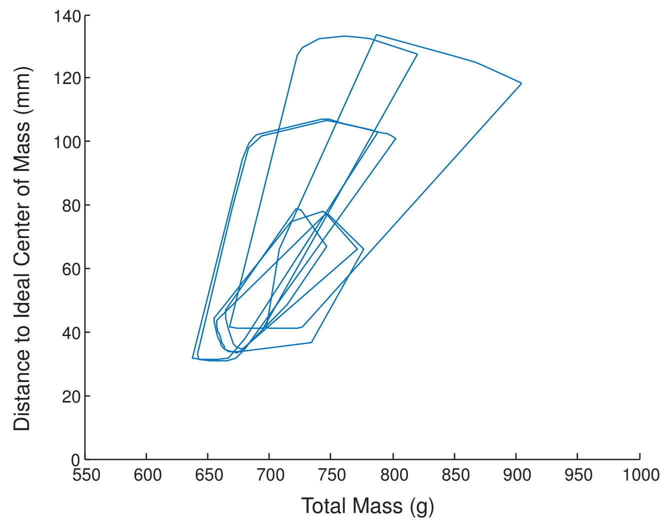


Figure 8.9: Formulation space for Concept 1.

quality, respectively. Each is an indication of improvement over the baseline design space ($k = 0$), and provide evidence that the exploration process has added value to the search.

A similar exploration process is performed for the remaining impact driver concepts. In each case, *P3* is used to find a suitable optimization problem formulation. The chosen formulations for each concept are numbered in Fig. 8.10. Three data points corresponding to the physical prototypes in Fig. 8.4 and to the Hitachi impact driver are also included in the plot. The asterisk marked with C1 represents the prototype for Concept 1, the asterisk marked with C4 represents the prototype for Concept 4, and the asterisk marked as H represents the 12V Hitachi impact driver. Since the featured models did not include the mass of the plastic casing, it was not included benchmark designs' mass in the plot. Also, these prototypes use a different motor than the one modeled, and therefore fall outside design spaces depicted for these concepts. In the figure, Concept 3 contains the largest portion of the s-Pareto frontier, and the predicted performance is significantly better than that of the existing prototypes. Perhaps if the BYU designers had access to this information which was provided by the exploration process, they would have made different decisions, and according the models, would have seen potentially better results.

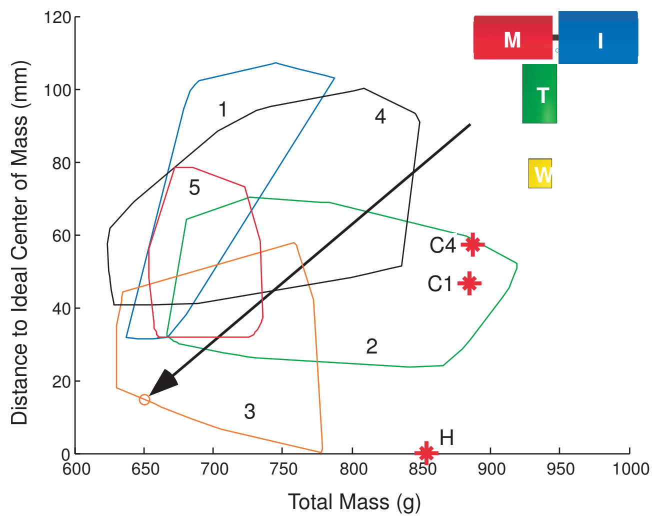


Figure 8.10: Comparison of five impact driver concepts. The asterisk marked ”H” represents the Hitachi WH10DFL model, and the asterisks marked ”C1” and ”C4” represent the prototypes from Fig. 8.4.

CHAPTER 9. CONCLUDING REMARKS

An optimization strategy has been presented that facilitates both convergence and divergence during conceptual design. Using this strategy, a computational search is not confined to the search space defined initially by an optimization problem formulation. Instead, a designer may search the formulation space, which has been defined as the union of all design variable and objective spaces identified by the designer as being valid and pragmatic problem formulations, to form the solution as he or she learns more about the design problem. This can open the door to early stage design exploration with computational assistance. As part of this strategy, a generic, vector/matrix-based, dynamic multiobjective optimization problem has been introduced, which allows the designer to easily modify and adapt the optimization problem as needed. Additionally, a set of exploration metrics has been developed, and three use scenarios for formulation space exploration have been identified. The theories and methods in this thesis have been demonstrated in two case studies involving the conceptual design of an antisubmarine warfare aircraft and an impact driver.

While the results of this thesis are promising, there are several avenues for improvement that can be made in future research.

- First, if the vision for the synergistic designer/computer design environment in Fig. 1.1 is to be realized, methods for inputting design concepts into the computer and automatically interpreting and parameterizing these concepts need to be developed or improved. The goal here is to let the human designer remain in a creative mind-set, while leaving the computer in place to execute numerical calculations.

- Second, the dynamic multiobjective optimization problem presented in Chapter 5 has a few limitations. It has been shown to be computationally inefficient when using gradient-based optimization algorithms. Moreover, it is required to separate design objects into independent and dependent objects, meaning, for example, that a dependent design constraint cannot be implemented directly as a design variable. Improvements in the dynamic optimization problem formulation should increase efficiency and allow for more flexibility.

- Third, as shown in the case studies in Chapter 8, proper visualization of optimization results can have a significant impact on formulation space exploration. An in-depth study of existing visualization methods and their application to formulation space exploration is warranted.

- Finally, several commercial optimization packages are currently available, including MATLAB, iSIGHT, and OptdesX to name a few. If optimization is to be widely used during conceptual design in industry, then software packages like these will have to "think outside the box" and begin to provide optimization methods that are more conducive to the conceptual design environment. Incorporating the ability to explore the formulation space would be a good starting point.

Improvements in these areas, coupled with the findings in this thesis, can facilitate the use of computational search methods throughout the design process. This will allow design engineers to make more informed design decisions, and ultimately, create better products.

REFERENCES

[1] Ulrich, K. T., and Eppinger, S. D., 2004. *Product Design and Development.*, third ed. McGraw-Hill/Irwin. 1, 3, 9, 13

[2] Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.-H., 2007. *Engineering Design: A Systematic Approach*. Springer. 1, 3

[3] Raymer, D. P., 1992. *Aircraft Design: A Conceptual Approach.*, 2 ed. American Institute of Aeronautics and Astronautics. 1, 29, 30

[4] Mattson, C. A., and Messac, A., 2003. "Concept selection using s-pareto frontiers." *AIAA Journal,* **41**(6), pp. 1190–1198. 1, 3, 22, 23

[5] Ishii, K., 1995. "Life-cycle engineering design." *Journal of Mechanical Design,* **117**, June, pp. 42–47. 3

[6] Homan, B. S., and Thornton, A. C., 1998. "Precision machine design assistant: A constraint-based tool for the design and evaluation of precision machine tool concepts." *Artifical Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM,* **12**(5), pp. 419–429. 3

[7] Wang, J., 2001. "Ranking engineering design concepts using a fuzzy outranking preference model." *Fuzzy Sets and Systems,* **119**, pp. 161–170. 3

[8] Mattson, C. A., and Messac, A., 2002. "A non-deterministic approach to concept selection using s-pareto frontiers." In *ASME IDETC/CIE2002, Montreal, Quebec, Canada, DETC2002/DAC-34125*. 3

[9] Olewnik, A. T., and Lewis, K. E., 2003. "On validating design decision methodologies." In *ASME DETC/DTM 2003, Chicago, Illinois, USA, DETC2003/DTM-48669*. 3

[10] Pugh, S., 1996. *Creating innovative products using total design: the living legacy of Stuart Pugh*. Addison-Wesley, Reading, Massachusetts. 3

[11] Brintrup, A. M., Ramsden, J., and Tiwari, A., 2007. "An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization." *Computers in Industry,* **58**, pp. 279–291. 3

[12] Brintrup, A. M., Ramsden, J., Takagi, H., and Tiwari, A., 2008. "Ergonomic chair design by fusing qualitative and quantitative criteria using interactive genetic algorithms." *IEEE Transactions on Evolutionary Computation,* **12**(3), June, pp. 343–354. 3

[13] Takagi, H., 2001. "Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation." *Proceedings of the IEEE,* **9**(9), September, pp. 1275–1296. 3

[14] Gong, D., and Yuan, J., 2011. "Large population size IGA with individuals fitness not assigned by user." *Applied Soft Computing,* **11**, pp. 936–945. 3

[15] Oduguwa, V., Roy, R., and Farrugia, D., 2007. "Development of a soft computing based framework for engineering design optimisation with quantitative and qualitative search spaces." *Applied Soft Computing,* **7**(1), January, pp. 166–188. 3

[16] Huber, M., Petersson, O., and Baier, H., 2008. "Knowledge-based modeling of manufacturing aspects in structural optimization problems." *Advanced Materials Research,* **43**, pp. 111–122. 3

[17] Barnum, G. J., and Mattson, C. A., 2010. "A computationally-assisted methodology for preference-guided conceptual design." *Journal of Mechancial Design,* **132**(12), p. 121003. 3

[18] Patel, J., and Campbell, M. I., 2010. "An approach to automate and optimize concept generation of sheet metal parts by topological and parametric decoupling." *Journal of Mechanical Design,* **132**, May, p. 051001. 3

[19] Mattson, C. A., Muller, A., and Messac, A., 2009. "Case studies in concept exploration and selection with s-pareto frontiers." *International Journal of Product Development,* **9**(1/2/3), pp. 32–59 Special Issue on Space Exploration and Design Optimization. 3

[20] Morino, L., Bernardini, G., and Mastroddi, F., 2006. "Multi-disciplinary optimization for the conceptual design of innovative aircraft configurations." *Computer Modeling in Engineering Sciences,* **13**(1), May, pp. 1–18. 3

[21] Hassan, R. A., and Crossley, R. A., 2002. "Multi-objective optimization of conceptual design of communication sattelites with a two-branch tournament genetic algorithm." In *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 3

[22] Qazi, M., and Linshu, H., 2005. "Rapid trajectory optimization using computational intelligence for guidance conceptual design of multistage space launch vehicles." In *AIAA Guidance, Navigation and Control Conference*. 3

[23] Dye, C., Staubach, J. B., Emmerson, D., and Jensen, C. G., 2007. "CAD-based parametric cross-section designer for gas turbine engine MDO applications." *Computer-Aided Design & Applications,* **4**(1-4), pp. 509–518. 3

[24] Davis, R., 2007. "Magic paper: Sketch-understanding research." *Computer,* **40**(9), sept., pp. 34 –41. 3

[25] Alvarado, C. J., and Davis, R., 2007. "Sketchread: a multi-domain sketch recognition engine." In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, ACM. 3

[26] Zeleznik, R., Miller, T., van Dam, A., Li, C., Tenneson, D., Maloney, C., and LaViola, J. J., 2008. "Applications and issues in pen-centric computing." *IEEE Multimedia,* **15**, pp. 14–21. 3

[27] LaViola, J. J., 2011. "Mathematical sketching: An approach to making dynamic illustrations." In *Sketch-based Interfaces and Modeling*, J. Jorge and F. Samavati, eds. Springer London, pp. 81–118. 3

[28] Landay, J. A., and Myers, B. A., 2001. "Sketching interfaces: toward more human interface design." *Computer,* **34**(3), mar, pp. 56 –64. 3

[29] Kuehmann, C. J., and Olson, G. B., 2009. "Computational materials design and engineering." *Materials Science and Technology,* **25**(4), pp. 472–478. 4

[30] Shelley, J. K., Giullian, N. C., and Jensen, C. G., 2007. "Incorporating computational fluid dynamics into the preliminary design cycle." *Computer-Aided Design & Applications,* **4**(1-4), pp. 235–245. 4

[31] Willcox, K., and Megretski, A., 2005. "Fourier series for accurate, stable, reduced-order models in large-scale applications." *SIAM Journal for Scientific Computing,* **42**(3), pp. 944–962. 4

[32] Wang, G. G., and Shan, S., 2007. "Review of metamodeling techniques in support of engineering design optimization." *Journal of Mechanical Design,* **129**, April, pp. 370–380. 4

[33] Shao, T., Krishnamurty, S., and Wilmes, G. C., 2007. "Preference-based surrogate modeling in engineering design." *AIAA Journal,* **45**(11), November, pp. 2688–2701. 4

[34] Allaire, D., and Willcox, K., 2010. "Surrogate modling for uncertainty assessment with application to aviation environmental system models." *AIAA Journal,* **48**(8), August, pp. 1791–1803. 4

[35] Galbally, D., Fidkowski, K., Willcox, K., and Ghattas, O., 2010. "Nonlinear model reduction for uncertainty quantification in large-scale inverse problems." *International Journal for Numerical Methods in Engineering,* **81**(12), March, pp. 1581–1608. 4

[36] Kiureghian, A. D., 1996. "Structural reliability methods for seismic safety assessment: A review." *Engineering Structures,* **18**(6), p. Engineering Structures. 4

[37] Frangopol, D. M., and Corotis, R. B., 1996. "Reliability-based structural system optimization: Stateof-the-art verses state-of-the-practice." In *Proceedings of the Twelfth Conference on Analysis and Computation*, pp. 67–78. 4

[38] Thanedar, P. B., and Kodiyalam, S., 1991. "Structural optimization using probabilistic constraints." In *AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference, number AIAA-91-0922-CP*. 4

[39] Melchers, R. E., 1999. *Structural Reliability: Analysis and Prediction*. Ellis Horwood Series in Civil Engineering. John Wiley & Sons. 4

[40] Parkinson, A., Sorensen, C., and Pourhassan, N., 1995. "A general approach for robust optimal design." *ASME Journal of Mechanical Design,* **115**, pp. 74–80. 4

[41] Chen, W., M.Wiecek, M., and Zhang, J., 1999. "Quality utility - a compromise programming approach to robust design." *ASME Journal of Mechanical Design,* **121**, June. 4, 20

[42] Chen, W., Sahai, A., Messac, A., and Sundaraj, G. J., 2000. "Exploring the effectiveness of physical programming in robust design." *ASME Journal of Mechanical Design,* **122**(2), pp. 155–163. 4

[43] Su, J., and Renaud, J. E., 1997. "Automatic differentiation in robust optimization." *AIAA Journal,* **35**(6), pp. 1072–1079. 4

[44] Taguchi, G., 1993. *Taguchi on Robust Technology Development: Bringing Quality Engineering Upstream*. ASME Press, New York. 4

[45] Messac, A., and Ismail-Yahaya, A., 2002. "Multiobjective robust design using physical programming." *Structural and Multidisciplinary Optimization,* **23**(5), pp. 357–371. 4

[46] Chen, W., and Wassenaar, H. J., 2001. "An approach to decision-based design." In *Proceedings of DETC01, ASME 2001 Design Engineering Technical Conference and Computers and Information in Engineering Conference, number DETC2001/DMT-21683, Pittsburgh, Penn.* 4

[47] DeVor, R. E., Chang, T. H., and Sutherland., J. W., 1992. *Statistical Quality Design and Control: Contemporary Concepts and Methods*. Prentice Hall, New Jersey, pp. 525–535. 4

[48] Koch, P. N., 2002. "Probabilistic design: Optimizing for six sigma quality." In *AIAA 43rd AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference, number AIAA20021471*. 4

[49] Scott, M. J., 2007. "Quantifying uncertainty in multicriteria concept selection methods." *Research in Engineering Design,* **17**, pp. 175–187. 4

[50] Arora, J. S., 2004. *Introduction to Optimum Design*. Elsevier Academic Press. 4

[51] Balling, R., 1999. "Design by shopping: A new paridigm?." In *Third World Congress of Structural and Multidisciplinary Optimization*, Vol. 1, pp. 295–297. 4, 32

[52] Stump, G., Lego, S., Yukish, M., Simpson, T., and Donndelinger, J., 2009. "Visual steering commands for trade space exploration: User-guided sampling with example." *Journal of Computing Information Science in Engineering,* **9**, pp. 044501–1–044501–10. 4, 26, 40

[53] Agate, J., de Weck, O., Sobieszczanski-Sobieski, J., Arendson, P., Morris, A., and Spieck, M., 2010. "MDO: Assessment and direction for advancement – an opinion of one international group." *Structural and Multidiscipilnary Optimization,* **40**(1), pp. 17–33. 4

[54] Finger, S., and Dixon, J. R., 1989. "A review of research in mechanical engineering design part I: descriptive, prescriptive, and computer-based modes of design processes." *Research in Engineering Design,* **1**(1), pp. 51–67. 5

[55] Evbuomwan, N. F. O., Sivaloganathan, S., and Jebb, A., 1996. "A survey of design philosophies, models, methods and systems." *Journal of Engineering Manufacture,* **210**(42), pp. 301–320. 5

[56] Yadav, O., Nepal, B. P., and Jain, R., 2007. "Managing product development process complexity and challenges: a state-of-the art review." *Journal of design research,* **6**(4), pp. 487–508. 5

[57] Markman, A. B., and Wood, K. L., 2009. *Tools for Innovation*. Oxford University Press. 5

[58] Linsey, J., and Viswanathan, V., 2011. "Enhancing engineering innovation through physical representation." In *Proceedings of 2011 NSF Engineering Research and Innovation Conference, Atlanta, Georgia*. 5

[59] Tumer, I. Y., Stone, R., Tumer, K., Oman, S., and Nix, A., 2011. "Collaborative research: Quantifying creativity in automated design through multi-agent coordination framework." In *Proceedings of 2011 NSF Engineering Research and Innovation Conference, Atlanta, Georgia*. 5

[60] Schunn, C., Wood, K., Cagan, J., Chan, J., Fu, K., and Kotovsky, K., 2011. "The effects of example distance and familiarity on conceptual ideation in engineering design: Initial results." In *Proceedings of 2011 NSF Engineering Research and Innovation Conference, Atlanta, Georgia*. 5

[61] Osborne, A., 1953. *Applied Imagination*. Charles Scribner and Sons, New York. 5

[62] Zwicky, F., 1948. *The Morphological Method of Analysis and Construction*. Wiley Interscience, New York Courant Anniversary Volume. 5

[63] Gordon, W. J. J., 1961. *Synectics*. Harper and Row, New York. 5

[64] Lubart, T., 2005. "How can computers be partners in the creative process: Classification and commentary on the special issue." *International Journal of Human-Computer Studies,* **63**, pp. 365–369. 5

[65] Kara, L. B., 2011. "Learning geometric knowledge from conceptual sketches and its utilization in shape design and optimization." In *Proceedings of 2011 NSF Engineering Research and Innovation Conference, Atlanta, Georgia*. 5

[66] Herold, J., and Stahovich, T. F., 2011. "Aligning speech and sketch modalities in multi-modal descriptions of engineering designs." In *Proceedings of 2011 NSF Engineering Research and Innovation Conference, Atlanta, Georgia*. 5

[67] Robertson, B. F., and Radcliffe, D. F., 2009. "Impact of cad tools on creative problem solving in engineering design." *Computer-Aided Design,* **41**(3), March, pp. 136–146. 5

[68] Mistree, F., and Allen, J. K., 1997. "Optimization in decision-based design." In *Decision-Based Design Workshop, Orlando, Florida* Position Paper. 5

[69] Simpson, T. W., and Martins, J. R. R. A., 2010. "The future of multidisciplinary design optimization: advancing the design of complex engineered systems." In *NSF Workshop Report, Fort Worth, Texas, September 16, 2010*. 5

[70] Miettinen, K. M., 1999. *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science. Kluwer Academic Publishers. 7

[71] Messac, A., and Mattson, C. A., 2002. "Generating well-distributed sets of pareto points for engineering design using physical programming." *Optimization and Engineering,* **3**(4), pp. 431–450. 7, 20

[72] Steuer, R. E., 1986. *Multiple Criteria Optimization, Theory Computations and Applications*. John Wiley & Sons, New York. 7, 20

[73] Belegundu, A., and Chandrupatla, T., 1999. *Optimization Concepts and Applications in Engineering*. Prentice Hall, New Jersey. 7

[74] Messac, A., and Mattson, C. A., 2004. "Normal constraint method with guarantee of even representation of complete pareto frontier." *AIAA Journal,* **42**(10), pp. 2101–2111. 8, 19, 34

[75] Fox, R. L., 1971. *Optimization Methods in Engineering Design*. Addison-Wesley. 10

[76] Shah, J. J., and Vargas-Hernandez, N., 2003. "Metrics for measuring ideation effectiveness." *Design Studies,* **24**, pp. 111–134. 18

[77] Halstead, M. H., 1977. *Elements of Software Science*. Elsevier North-Holland, Inc., Amsterdam. 21

[78] McCabe, T. J., 1976. "A complexity measure." *IEEE Transactions on Software Engineering,* **2**(4), December, pp. 308–320. 21

[79] Mattson, C. A., and Messac, A., 2005. "Pareto frontier based concept selection under uncertainty with visualization." *Optimization and Engineering,* **6**(1), pp. 85–115. 23

[80] Das, I., and Dennis, J. E., 1998. "Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems." *SIAM Journal on Optimization,* **8**(3), August, pp. 631–657. 24

[81] Messac, A., Ismail-Yahaya, A., and Mattson, C. A., 2003. "The normalized normal constraint method for generating the pareto frontier." *Structural Multidisciplinary Optimization,* **25**, pp. 86–98. 24

[82] Blasco, X., Herrero, J. M., Sanchis, J., and Martínez, M., 2008. "A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization." *Information Sciences,* **178**(20), October, pp. 3908–3924. 40

[83] Huang, C. H., and Bloebaum, C. L., 2004. "Visualization as a solution aid for multi-objective concurrent subspace optimization in a mulidisciplinary design environment." In *Proceedings of 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 40

[84] Jones, C. V., 1996. *Visualization and Optimization*. Springer. 40

APPENDIX A. PSUEDO CODES

<u>Standard Multiobjective Optimization Code (P1)</u>

```
Define Design Variable Limits...
Define Design Parameter Values...
Construct x0
Construct xL
Construct xU
Construct P
Call [x*, mu*] = optimize(x0, xL, xU, P)

function [objectives] = objectiveFunction(x, P)
Call [outputs] = model(x, P)
Calculate objectives

function [g, h] = constraintFunction(x, P)
Call [outputs] = model(x, P)
Define Equality Constraint Values...
Calculate h...
Define Inequality Constraint Values...
Calculate g...

function [outputs] = model(x, P)
Extract x...
Extract P...
Calculate outputs...
```

<u>Dynamic Multiobjective Optimization Code (P2)</u>

```
Define Independent Design Object Limits...
Define Dependent Design Object Limits...
Construct y0
Construct yL
Construct yU
Construct zL
Construct zU
Define w...
Call [x*] = optimize(y0, yL, yU, zL, zU, w)

function [objectives] = objectiveFunction(y, w)
Call [z] = model(y)
Calculate x = [y;z]
Calculate objectives = w*x

function [constraints] = constraintFunction(y, zL, zU)
Call [z] = model(y)
Calculate constraints = [zL-z; z-zU]

function [z] = model(y)
Extract y...
Calculate z...
```