2010-07-02

A Computationally-assisted Methodology for Rapid Exploration of Design Possibilities in Conceptual Design

Garrett J. Barnum
Brigham Young University - Provo

A COMPUTATIONALLY-ASSISTED METHODOLOGY FOR RAPID

EXPLORATION OF DESIGN POSSIBILITIES

IN CONCEPTUAL DESIGN

Garrett J. Barnum

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
Robert H. Todd

Department of Mechanical Engineering

Brigham Young University

August 2010

ABSTRACT

A COMPUTATIONALLY-ASSISTED METHODOLOGY FOR RAPID

EXPLORATION OF DESIGN POSSIBILITIES

IN CONCEPTUAL DESIGN

Garrett J. Barnum

Department of Mechanical Engineering

Master of Science

One of the most important decisions in the product development process is the selection of a promising design concept because of the large influence it has on the final product. A thorough search for the best design is a significant challenge to designers, who are trying to balance the objective and subjective performance of the designs they create. In this thesis, a computationally-assisted design methodology is developed and used in the early stages of design to more thoroughly search for designs that perform well according to objective physics-based models *and* subjective designer-specific preference-based models. The method presented herein uses an initial pool of user-created designs that is parameterized and used in a numerical search that recombines design features to form new designs in a semi-automated way. Designs are then evaluated quantitatively by objective performance calculations and evaluated qualitatively by human designers. Designer preference is interactively gathered when visual representations of new computer-created designs are presented to the designer for subjective evaluation. A mathematical model is then formed using statistical probability methods to approximate the designer's preference and incrementally updated after the designer subjectively evaluates a new set of designs at each iteration of the automated search process. The methodology uses a multiobjective approach to search for optimally performing designs, treating both the physics-based models *and* the preference-based models as objectives. The methodology couples the speed of computational searches with the ability of human designers to subjectively evaluate unmodeled objectives. The method is demonstrated with two product examples to find optimal designs that designers may not have otherwise discovered among the vast number of possible combinations of features. The proposed methodology brings the ability to search for and find numerous, optimal solutions across a wide solution space, in an efficient and human-centered way, and does so in the early stages of design.

Keywords: conceptual design, concept generation automation, multiobjective optimization, preference capture, Garrett Barnum

ACKNOWLEDGMENTS

I would like to acknowledge all those who have helped and supported me throughout this research. I would like to thank my advisor and friend, Dr. Christopher A. Mattson, for all his help and guidance in the direction of this thesis. His insight and previous work gave me focus and a foundation upon which I built much of my research. I would also like to thank my committee members, Dr. Spencer P. Magleby and Dr. Robert Todd, for their validation and advice in all aspects of my life. Jacob Montierth, of the Ford Motor Company, also provided input and validation of the vehicle platform design example. I am grateful to the Department of Mechanical Engineering of Brigham Young University that provided funding for this work. I acknowledge the constant blessing of Divine inspiration and assurance in my studies, this research, and all aspects of my life. Above all, I express my appreciation to my wife Kricket and my children, Jenna, Cooper, Ryder, and Trey (who was born 8 days before my thesis defense), for their unending love, support, and encouragement to persevere, and their reminder that "I can do hard things."

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

c	Design chromosome
F_m	Numerical Morphological Matrix
F_c	Matrix of lower and upper bounds for continuous design variables
$\hat{f}$	Probability density estimate
G	Number of generations
g	Vector of inequality constraints
h	Vector of equality constraints
μ	Vector of design objectives
μ^{phys}	Vector of physics-based design objeectives
μ^{pref}	Vector of preference-based design objeectives
N	Number of designs in population/generation
n_V	Number of designs presented to the designer for evaluation
n_F	Percentage of designs not selected randomly to be presented to the designer
p	Number of iterations of the Preference-guided Search, Steps 6-10
r_R	Percentage of designs selected randomly to be presented to the designer
σ	Standard deviation
x_d	Vector of discrete design variables
x_c	Vector of continuous design variables

Subscripts, superscripts, and other indicators

$n_{[\ ]}$	indicates the number of [ ]
$[\ ]_l$	indicates the lower limit of [ ]
$[\ ]_u$	indicates the upper limit of [ ]

GLOSSARY

designer A human designer or development engineer who contributes to finding solutions and developing products [1]

design requirement A condition or customer need that a product, concept, or design must satisfy

functional specification A metric and a value, providing a measurable level of performance corresponding to a design requirement [2]

concept A *human-generated* design; consisting of a partial or complete description of the form, functions, features, and other attributes which have the intent of meeting the design requirements of a product

design A *computer-generated* design; consisting of a set of features and attributes which have the intent of meeting the design requirements of a product

function An abstract formulation of the overall task needed to meet the product requirements [1]

subfunction Smaller divisions of the overall function; corresponds to subtasks [1]

feature An embodiment of ideas that accomplishes specific subfunctions

conceptual design One of the early phases of product development that includes identifying target market needs, generating and evaluating alternative product concepts, and selecting one or more concepts for further development [2]

detailed design The phase of product development which further develops the concept identified in previous phases through the use of computational tools which use predictive models to analyze and incrementally improve a design's performance in an effort to finalize the the final geometry, materials, and tolerances [2]

physics-based model A quantitative parametric model, representing natural laws and theories, however simplified, used to approximate or predict the physical performance of a design according to specific metrics, calculated as a function of a set of input variables

preference-based model A *quantitative* model of the *qualitative* preference of a designer, approximating his/her like or dislike for a design as described by a set of input variables

CHAPTER 1. INTRODUCTION

1.1 Problem Introduction

Among the most important decisions in the product development process is one that marks the end of the conceptual design phase – the selection of the most promising design concept – which will be fully developed in the remaining phases of the development process. While there are various effective methodologies to assist the designer in identifying the best performing concepts within a given set [1–5], these methods are limited by the quality and quantity of the set of concepts under consideration. The quality of that set is partially determined by the level of creativity, intuition, and experience of the design team, while the quantity is primarily determined by the amount of time and effort given to concept generation activities. Unfortunately, the abstract, ambiguous, and open-ended nature of conceptual design makes it impractical to generate and consider all, or even a majority of the possible concepts using manual methods.

The methodology presented herein, as well as other conceptual design automation research [6–12], uses *computational methods*, which are traditionally used in the design embodiment and detailed design phases of the development process, to automatically and rapidly search through the large conceptual design space in a more thorough and efficient way than can be done manually. However, quantitative, predictive models are required in order to automatically evaluate performance of new concepts, and in the early phases of design, there are often subjective decisions made which are difficult to quantify. It would be advantageous to have a way to model these subjective decisions and use them in a computational search for two main reasons: (1) a significantly higher number of concepts could be automatically evaluated in a rapid manner, and (2) by more fully exploring the design possibilities in an automated way, there is an increased chance to find better performing, more preferred designs.

Throughout this thesis, all predictive models will be classified as one of two types, physics-based models or preference-based models, which are defined here.

A Physics-based Model is a quantitative parametric model, representing natural laws and theories, however simplified, used to approximate or predict the physical performance of a design according to specific metrics, calculated as a function of a set of input variables.

A Preference-based Model is a *quantitative* model of the *qualitative* preference of a designer, approximating his/her like or dislike for a design as described by a set of input variables.

Traditional physics-based performance can be modeled by equations, calculated using design input variables, and can be numerically optimized. However, the existence of subjective decisions implies that there are important aspects of performance that are unmodeled by physics-based models, but *are* modeled in the mind and intuition of the experienced designer, and these subjective decisions are influential during conceptual design. Completely excluding humans from an automated evaluation process would ignore the importance of human subjectivity in decisions made during the conceptual design phase.

1.2 Research Context

The design methodology presented herein is intended to be used as a decision-making assist/tool during conceptual design, after an initial set of manually generated concepts has been generated, but before the selection of a final concept. The design methodology automates the traditionally manual process of combining and recombining features from an initial set of concepts, but also performs statistical parameter studies, uses physics-based models, uses preference-based models when available, and allows the designer to introduce new, human-generated concept features which were not present in the initial set of concepts, all in an effort to guide the search for better performing combinations of the human-generated ideas/features.

Figure 1.1 compares a conventional conceptual design process, on the left [4], with the proposed computationally-assisted design methodology in the conceptual design process on the right [13]. Both begin with a set of manually, human-generated concepts. The conventional process gradually converges on the best concepts, using human-based judgment and preferences to creatively generate, combine, and evaluate the concepts of each subsequent iteration until the final concept emerges.

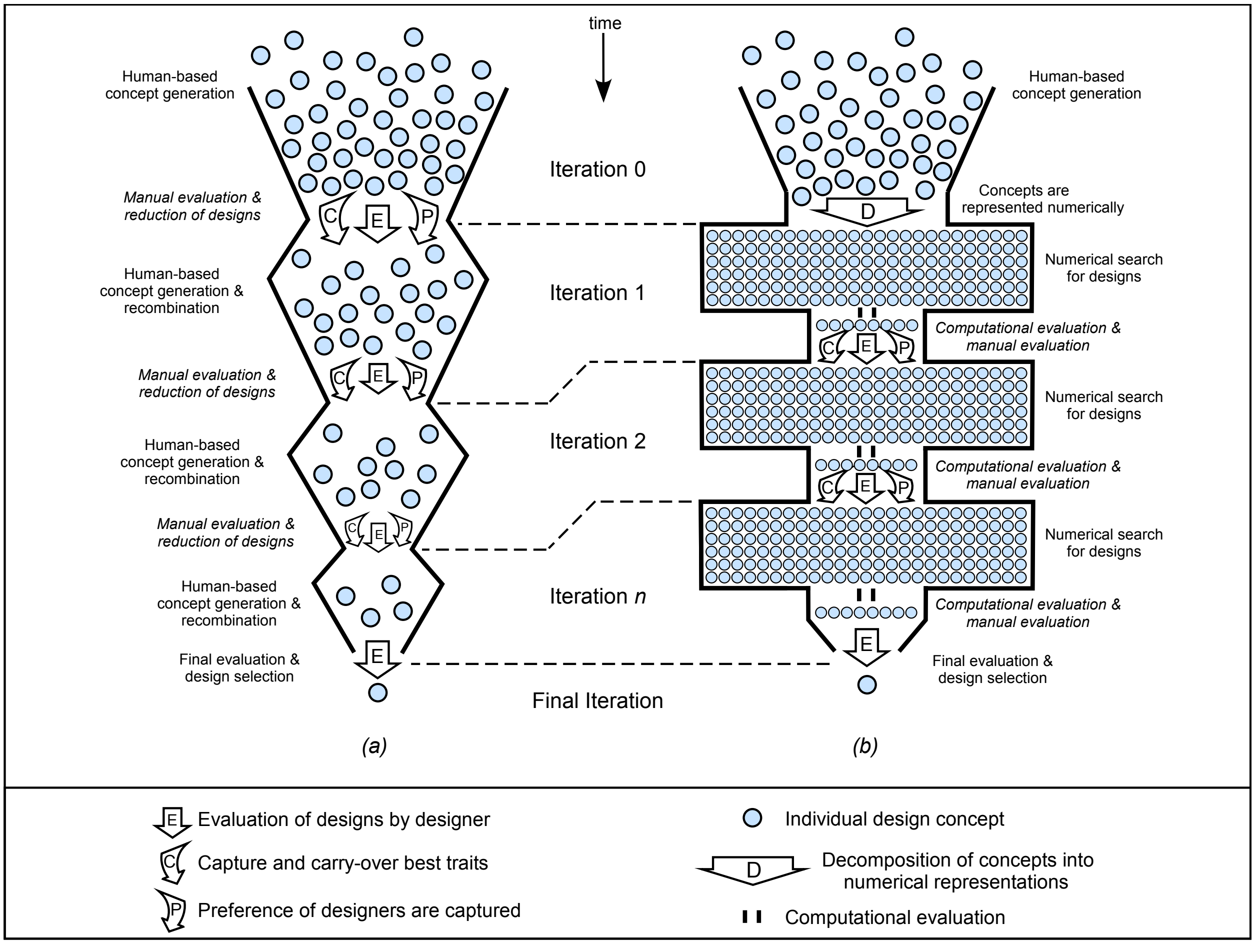


Figure 1.1: A comparison of (a) a conventional conceptual design process [4] and (b) the proposed computationally-assisted design methodology in a conceptual design process [13].

The proposed design methodology, in the process on the right of Figure 1.1, partially automates the steps of conceptual design by using a numerical search to automatically form new combinations of features, automatically evaluate those designs, and find the best performing designs of each iteration, which are then presented to designers for manual evaluation. In this way, the best concept traits and designer preferences are captured, incorporated into the numerical search, and used to guide the search for designs in the next iteration. By automating the search, the computationally-assisted design methodology rapidly explores and evaluates thousands of designs within minutes, thus expanding the search for the best designs beyond traditional, manual methods.

1.3 Objectives

The main objective of this thesis is to develop a design methodology that uses computational power to assist designers in the search for optimally performing designs according to physics-based models *and* preference-based models. To accomplish the development of this methodology, the main objective is broken down into the following sub-objectives:

1. Develop a method that uses numerical representations of human-generated design concepts to find *optimal designs* according to physics-based performance models

2. Develop a method to capture designer *preferences*, form a quantitative preference-based performance model, and use it to find more preferred designs

3. Develop a multiobjective search strategy that concurrently accomplishes the previous sub-objectives by finding and evaluating *significantly more* number of designs than could otherwise be done manually; identifying designs that would not have been found otherwise

The methodology presented in this thesis addresses the following challenges which are associated with accomplishing these objectives.

- Creating numerical representations of design concepts, or partial concepts, that may only be described with sketches

- Evaluating performance of concepts that require completely different models to evaluate performance

- Capturing and representing the subjective preferences of a designer in a mathematical model

- Performing a multiobjective search for optimal designs, guided by a constantly changing model of human preference

The development of the design methodology in this thesis research has resulted in several contributions relating to the objectives and challenges listed above. Initial research efforts resulted in a conference publication [13] and the development of the methodology to decompose and represent numerically, or parameterize, human-generated design concepts to enable the computational exploration of morphological charts for optimally performing designs according to physics-based performance models. It also resulted in the development of the design methodology and search strategy to find and evaluate significantly more number of designs, on the order of thousands of designs in a few minutes, than could be done in a manual conceptual design process. The publication formed a foundation upon which the next paper was built.

Additional research resulted in a conference paper which has been accepted for publication on the use of the computationally-assisted design methodology for preference-guided conceptual design [14]. This work developed the portion of the design methodology which captures designer preferences, forms a quantitative preference-based performance model, and incorporates the model into a multiobjective search for better designs, including both physics-based objectives *and* preference-based objectives.

Outline

The remaining chapters of this thesis are organized as follows: Chapter 2 provides a literature survey related to conceptual design automation, preference capture, and several other foundational technical topics. Chapter 3 introduces the new computationally-assisted design methodology graphically and provides a very brief overview of its steps. Chapter 4 explains the first half of the steps, identified as the *Numerical Optimization Search Strategy*. Chapter 5 explains the remaining steps of the methodology, identified as the *Preference-guided Search*. Chapter 6 demonstrates the use of the new design methodology with two product examples, a table design problem and a vehicle design problem, and discusses the results. In the final chapter of this work, conclusions are drawn and recommendations are made for continuing work that could be completed in this area of research. Potential applications and limitations of the method are also discussed.

CHAPTER 2. LITERATURE SURVEY

The first section of this chapter reviews current research in conceptual design automation, and the important developments to quantify designs during conceptual design in order to automate the search and creation of new designs. The second section reviews recent advancements to capture, model, and incorporate human preference into computational methods. The third section provides background on several other technical topics which are foundational to the development of the computationally-assisted design methodology, presented in this thesis.

2.1 Conceptual Design Automation

There are two basic strategies in conceptual design; convergent and divergent [1,3,15]. The convergent strategy seeks a promising concept as efficiently as possible. That is, in the quickest amount of time, having spent the least amount of resources. One drawback to the convergent method is that it may not identify the best performing concepts. The divergent strategy, on the other hand, seeks to identify the most promising concepts by first exploring numerous, diverse, possibilities. As such, the divergent strategy is effective at identifying a diversity of solutions across a wide solution space. A drawback to this strategy is the potentially large amount of time it can take to generate and evaluate a sufficient range of solutions, resulting in the early rejection of ideas that are not fully explored and evaluated. One purpose of concept generation automation research [6–12], is to utilize computational power to generate and search through a large, diverse population of concepts in a way that is as thorough as the divergent strategy, but as efficiently as the convergent strategy.

Design Decomposition and Morphological Charts

Decomposition methods and morphological charts are reviewed because they are a means to form and organize designs. Chapter 4 explains how the new computationally-assisted design methodology presented in this thesis uses this organization method to create numerical representations of designs.

Within the discipline of design engineering, it is common practice to divide, or decompose, large problems into smaller, more manageable problems [2]. The most common decomposition methodologies map functional requirements to physical design parameters [16]. The foundational "black box" definition of design decomposition presented by Pahl and Beitz [1] involves decomposing the main product function into sub-functions. A function structure is created by determining the sequence and relationships of sub-functions and identifying their input and output flows through each sub-function. Pahl and Beitz also suggest five general function types and three types of flows. The Russian methodology of TRIZ identifies 39 functional descriptions for all mechanical design functions [17]. By using a framework of functional structures, designers can consider alternative means to accomplish each function of a design. Designers can organize the alternative functions of a design in a combination table, also called a morphological chart.

Morphological charts are used to organize the parts of a decomposed design problem to aid in forming alternative combinations of solutions. Figure 2.1 shows a generic morphological chart. Each row in the chart contains alternative solutions for the functions in each row. Designers manually select a combination of solutions to form a complete design. When the combination process is manually performed, the range and diversity of designs formed is limited by the ability of designers to manage the alternatives in the morphological chart, or by the ability and patience of designers to enumerate many, or all combinations, which can easily be millions of designs. The proposed approach in this thesis computationally forms and organizes combinations, thus increasing the number of combinations that can be considered.

Concept Generation Automation

When considering anything but a small morphological chart it can be overwhelming for a designer to manually organize the many possible combinations into complete designs, and to manage very many alternative designs, in a way that facilitates quick and effective evaluation. Also, as the number of functions and alternative solutions for the functions increase, it quickly becomes beyond the capacity of human designers to understand how the overall concept is affected by different combinations of functional solutions within a reasonable amount of time as the number of

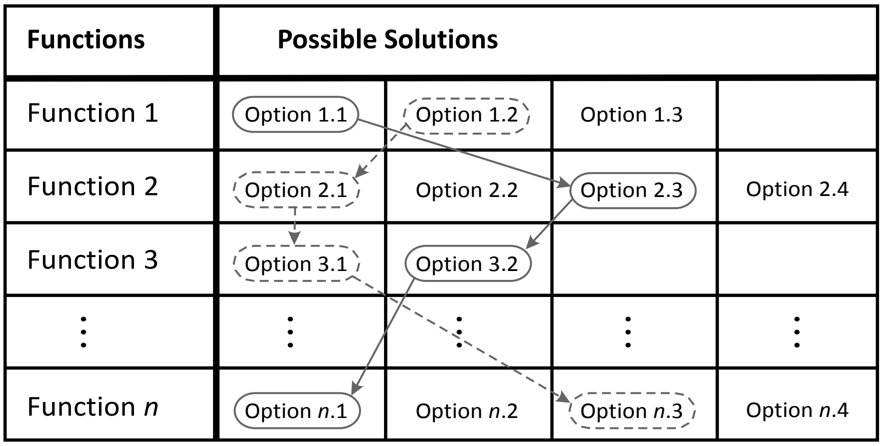


Figure 2.1: A morphological chart showing possible solutions for the decomposed functions of a design problem. Two combinations of solutions are shown.

unique design combinations N in a morphological chart is $N = m_1 \cdot m_2 \cdot m_3 \cdots m_n$ where m_n is the number of possible solutions for the n-th function in each row of a morphological chart [1]. To handle the vast amount of data, software programs have been developed that use digital morphological charts [6], allowing designers to visually select options from the chart. This may help in forming new designs, but does not address the need to manage the vast number of new designs and the trade-offs in performance that come from selecting different combinations of features.

Design repositories help manage the vast amount of design knowledge embedded in individual designs, by digitally archiving solutions from existing products based on the functions they perform. Archived product information is searched to find similar solutions that are then suggested for a new product with the same functions [7–9]. Currently, researchers at Missouri University of Science and Technology, in collaboration with the University of Texas at Austin, have produced an automated software tool that draws on a design repository with stored design information from over 100 consumer products to produce new designs [10, 11]. While the use of design repositories draws on a vast number of archived solutions, a manual screening process is still required by designers to select the solutions which are applicable to the specific situation at hand.

There exists several major issues that exist when design features are automatically combined to generate a large population of diverse concept variants; one issue is compatibility. Understanding the interactions between design features in design decomposition are very important according to Pimmler and Eppinger [18]. This is because of the complexity of the feature interactions and the impact they have on the formation of accurate product architecture. The design attribute encapsulation (DAE) method groups design attributes according to common compatibility traits [19]. This results in rules and constraints that only allow the creation of feasible combinations of design attributes. The internet-based ProDefine system can allow a virtual design team to input "goals", "means to achieve goals", and other feasibility rules to generate variants as new combinations of means and goals [20, 21]. Allen and Carlson-Skalak place importance on modularity in determining which combinations of design features are compatible [22]. They find that improvements for new products can be found by analyzing the modularity and function structures of a company's current products. These important approaches help facilitate the combinations of solutions, however, new concepts are still limited to the combination of solution options that designers manually select and evaluate.

Other conceptual design automation research by Hutcheson et al., shows that genetic algorithms can be used to select multiple designs for detailed evaluation based on quantitative objectives formulated during conceptual design [12]. Other work has shown the use of pattern search algorithms for efficient two and three dimensional packaging, layout, and routing problems [23, 24]. While these methods form designs automatically and use traditional numerical optimization to find better performing designs, human subjectivity and preference, which are present and very influential during conceptual design, are excluded from the search and evaluation process completely. The designer preference capture and incorporation method developed in this thesis is explained in Chapter 5.

2.2 Preference Capture

Related to the preference capture methods presented in Chapter 5 of this thesis, recent advances in computational power have given rise to many practical uses in machine learning techniques, attempting to recognize complex patterns and make intelligent decisions based on data [25], and in the area of interactive evolutionary computation (IEC), all of which incorporate human evaluation into the numerical search process.

Neural Networks and Machine Learning

Design experience and knowledge are very valuable but difficult to transfer to other designers or from one product to the next. A major focus of research in the areas of neural networks and machine learning is to automatically learn to recognize complex patterns and make decisions based on preexisting data [25]. Artificial intelligence methods such as these have been used in creative and subjective applications, such as in the creation of art [26] and to recommend music [27] based on prior learning.

Similarly, the decisions that designers make in the early phases of the design process involve a complex and large set of data and variables that are used to make quantitative *and* qualitative evaluations. While machine learning methods are good at finding patterns in complex data that would not be recognizable to humans, they generally require large sets of training data to "learn" from and to develop the algorithms that can make accurate future decisions.

Interactive Computation

Other fields of design research have done work to use interactive human input to help make decisions. Michalek and Papalambros demonstrate the use of visual representations of architectural layouts and enable designers to have a level of interactive optimization [28]. This is shown to be useful in architectural layout design because of the subjectivity related to the performance of floorplan designs [29].

Recent research to use evolutionary computation (EC) methods in engineering design acknowledges that there are potential gains from using subjective human evaluation to guide optimization towards better solutions, especially when the problem is less well-defined such as during conceptual design [30]. Takagi [31] provides an excellent survey of Interactive Evolutionary Computation (IEC) research following the definition of IEC as "the technology that EC optimizes the target systems based on subjective human evaluation as fitness values for system outputs" (See Figure 2.2). Takagi surveys over 250 papers on IEC research with applications in the fields of graphic arts and animation, music, editorial design, industrial design, facial image generation, speech processing and synthesis, hearing aid fitting, virtual reality, media database retrieval, data mining, image processing, control and robotics, food industry, geophysics, education, entertain-

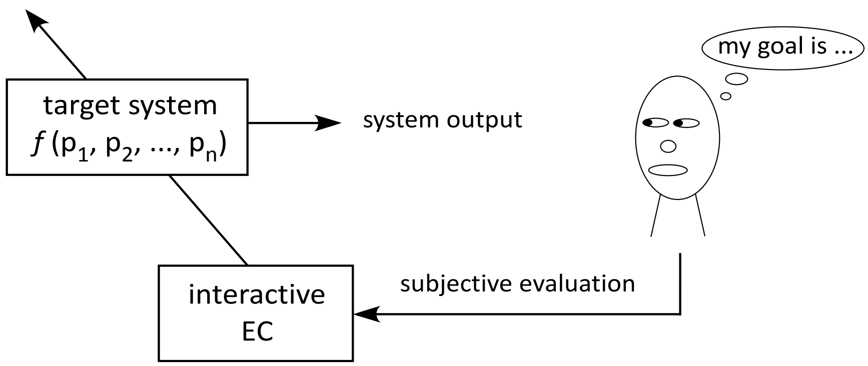


Figure 2.2: A general IEC (Interactive Evolutionary Computation) system: system optimization based on subjective evaluation [31].

ment, social systems, and so on. While initial research was primarily concerned with subjective improvements of solutions in artistic fields, other researchers have applied IEC to engineering and other practical fields. The primary challenges of IEC research include human fatigue, prediction of human preference through fitness values, and the nature of the active human interface used for intervention.

Similarly, the subjective decisions that designers make in the early phases of the design process involve a complex mix of quantitative *and* qualitative evaluations. Therefore, in an effort to model the subjective preferences of designers and increase the probability that automatically formed designs will be more preferred by designers, the method presented in the following chapters actively builds a quantitative model of a designer's subjective preference and interactively updates that model throughout the process. Incorporating preference into the automated numerical search for better performing designs is one of the major contributions of this thesis.

2.3 Other Foundational Technical Topics

A summary of probability density estimation and data smoothing techniques is also given as it relates to the methods developed in this thesis to actively capture designer preference and form quantitative and predictive, preference-based models. Background on multiobjective optimization is also given as background for the numerical search strategy used in this work. This work does not present advancements in these topics, but their use is foundational to the development of the computationally-assisted design methodology in this thesis.

Statistical Probability Density Estimation

The foundational statistical concept used in this thesis for the creation of a predictive preference-based model is the *probability density function*, which is defined as the function f that gives the probability P that an event X will occur within certain bounds (a,b), as shown below in Equation 2.1 [32].

$$P(a < X < b) = \int_a^b f(x)dx \qquad \text{for all } a < b \tag{2.1}$$

A *probability density estimate* is an estimate of the probability density function $\hat{f}$ constructed from a set of observed data [32]. A very simple form of a probability density estimate is a histogram, where the frequency of occurrences within intervals, called bins, are represented by the height of bars on a bar chart. For a histogram, the probability estimate $\hat{f}$ for a new point x is the proportion of the sample X_i within the same bin [32], as shown below in Equation 2.2,

$$\hat{f}(x) = \frac{1}{nh}(\text{no. of } X_i \text{ in same bin as } x) \tag{2.2}$$

where n is the number of bins and h is the width of the bins. Variable h is usually called the *smoothing parameter* or *bandwidth*, because of its control over the amount of "smoothing" that is applied to the data. The histograms in Figure 2.3 show how the same set of data can look very different when a different bin width, or smoothing parameter, is used [33].

Another type of probability density estimate is the *kernel density estimator*, which by definition is

$$\hat{f}(x;h) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{2.3}$$

where the kernel K is a distribution satisfying $\int K(x)dx = 1$ (i.e. a normal distribution), and h is the *smoothing parameter* [32]. This type of density estimate is illustrated in Figure 2.4 where the individual kernels (the small distribution curves placed over each data point) are summed up to form the density estimate $\hat{f}$ [33]. As with the histogram, the smoothness of the curve is determined by the bandwidth, or smoothing parameter h, of the individual kernels. Figure 2.5 shows how changing the smoothing parameter can introduce unwanted variation in the curve or over-smooth the curve [33]. The *normal optimal smoothing* method is one of the most common, and most

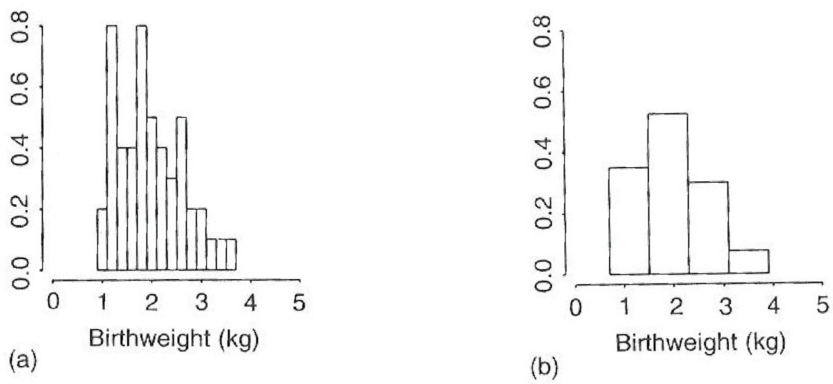


Figure 2.3: Two histograms of the same data with different bin widths, or smoothing parameters [33].

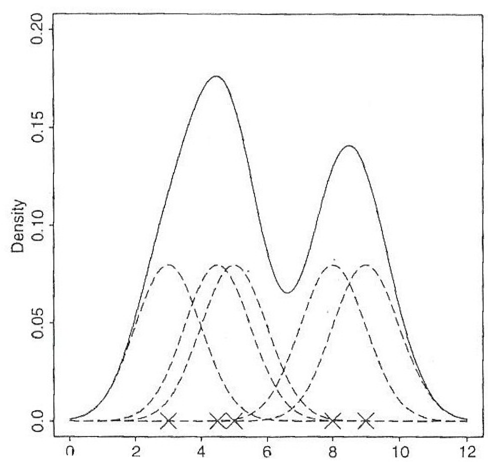


Figure 2.4: A kernel density estimate [33].

effective methods to choose a smoothing parameter [34]. Assuming the kernel K is a normal density, the smoothing parameter h is calculated as

$$h = \left(\frac{4}{3n}\right)^{1/5} \sigma \tag{2.4}$$

where σ is the standard deviation of the distribution [34].

It is effective to use methods such as these to empirically form the probability density estimates when the nature of the data is unknown or is not suspected to follow any common continuous

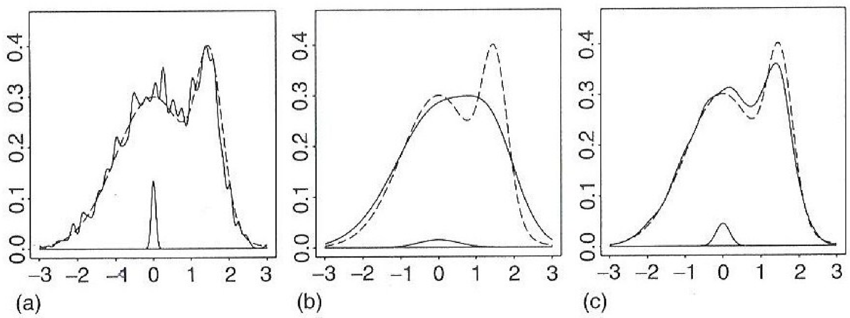


Figure 2.5: Kernel density estimates with different smoothing parameters [33].

distributions (such as a exponential, lognormal, or Poisson distributions), such as will be shown in Chapter 5 when building a quantitative model of a human designer's preference.

Multiobjective Optimization

Many decisions that designers make in the early phases of the design process involve trade-offs between quantitative *and* qualitative design aspects. In the context of the conceptual design methodology of this thesis, there are also trade-offs between physics-based objectives *and* preference-based objectives, which are handled through multiobjective optimization methods, as well as trade-offs made when a designer subjectively evaluates designs. The design objectives $(\mu_1, \mu_2, ..., \mu_{n_\mu})$ are competing – if one improves, the other gets worse – resulting in not one unique solution, but a set of solutions with varying degrees of performance in each objective [35–43]. Identifying the best designs requires the identification of a Pareto frontier – a set of nondominated optimal solutions. Figure 2.6 shows a feasible design space with two objectives, μ_1 and μ_2, and a Pareto frontier. All solutions along the frontier are said to be *Pareto optimal* – no other designs better satisfy *all* design objectives [44–47].

A generic multiobjective optimization problem (MOP) formulation yielding a set of optimal solutions – those belonging to the Pareto frontier – is presented as follows:

Problem 1: Generic multiobjective optimization problem statement

$$\min_x \{\mu_1(x), \mu_2(x), ..., \mu_{n_\mu}(x)\} \qquad (n_\mu \geq 2) \tag{2.5}$$

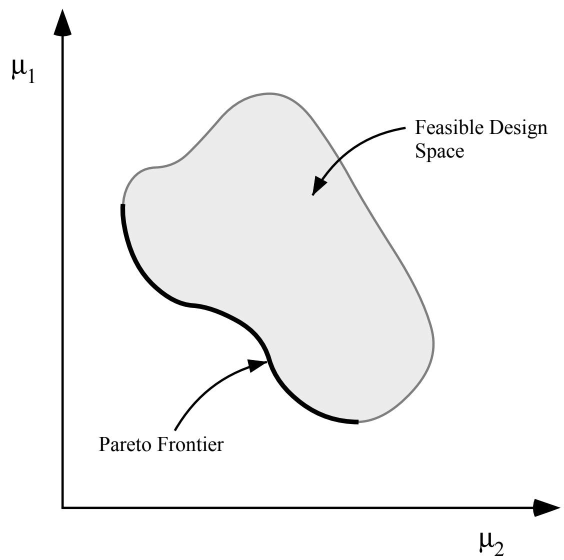


Figure 2.6: A graphic of a feasible design space and the Pareto Frontier.

subject to:

$$g_q(x) \leq 0 \quad \forall q \in \{1,...,n_g\} \tag{2.6}$$

$$h_v(x) = 0 \quad \forall v \in \{1,...,n_h\} \tag{2.7}$$

$$x_{jl} \leq x_j \leq x_{ju} \quad \forall j \in \{1,...,n_x\} \tag{2.8}$$

where μ_i denotes the i-th generic design objective; g is a vector of inequality constraints; h is a vector of equality constraints; x is a vector of design variables; and the design variables are bound by their lower *(l)* and upper *(u)* limits shown in Equation 2.8.

Maximin Fitness Function

Regarding the numerical search used in the conceptual design methodology of this thesis, the goal of the search is to find a diverse set of designs that are representative of the entire design space, finding designs that designers would not have otherwise found through manual methods. For that reason the Maximin fitness function [35, 48] is introduced. The Maximin fitness function is derived from the definition of dominance, and is represented mathematically in Equation 2.9. The Maximin fitness of design i, within the set of all other designs j, is:

$$f^i_{\text{maximin}} = \max_{j \neq i} \left(\min_{\mu_n} \left(f^i_{\mu_n} - f^j_{\mu_n} \right) \right) \tag{2.9}$$

where $f^i_{\mu_n}$ is the value of the μ_n-th objective for design i, and $f^j_{\mu_n}$ is the value of the μ_n-th objective for all other designs within the set j.

The Maximin fitness function penalizes clustering of non-dominated designs, which forces the designs to spread out across the design space and the Pareto frontier. The Maximin fitness function is a minimizing function, so the designs with lower fitness values are better, and in fact, positive values indicate dominated designs and negative values indicate non-dominated designs.

Genetic Algorithms

Evolutionary algorithms, such as genetic algorithms, are well suited for computational searches performed early in the design process. Evolutionary algorithms generate a population of individuals/solutions in each iteration, all converging on a single optimum, or potentially converging on multiple optimum solutions [49]. A population of solutions is also helpful to explore the complex design space of a multi-objective optimization problem when multiple trade-off solutions are required.

CHAPTER 3. OVERVIEW OF DESIGN METHODOLOGY

An overview is given of the new computationally-assisted design methodology in this chapter, and further explained in Chapters 4 and 5. In its basic form, the new design methodology can be described graphically in Figure 3.1. As shown, the methodology has eleven major steps spanning from the earliest design activities – *Define Problem & Design Requirements* – to one of the final conceptual design activities – *Selection of Final Design(s)*. It is important to note, now, that the designer using the proposed method would need to carry out Steps 1 through 3 using any traditional design method. Each of the remaining steps explain methods to overcome the challenges in accomplishing the objectives outlined in Chapter 1.

Step 4 specifically addresses the objective of representing manually generated designs numerically – a step called *parameterization* – which enables the designs to be used in the numerical search of the design methodology. Step 5 addresses the challenge of defining mathematical performance models to calculate physics-based *and* preference-based design performance. Step 6 addresses the challenge of using a constantly changing preference-based model by the formation of an optimization problem statement on the first cycle, and updating that problem statement on all subsequent cycles. Step 7 performs the numerical search for better performing designs according to the current optimization problem statement. Step 8 selects a subset of designs to present to the designer, and visually presents the designs for the manual evaluation performed by the designer in Step 9. Step 10 uses the designs selected in Step 9 to address the objective to form a quantitative preference-based model. This preference model is then carried forward and used in the optimization in Step 7 of the next cycle, as indicated in the figure by the broken line leading into Step 7. After initial setup, Steps 6 through 10 can be executed automatically. In this way, the operations of the numerical search can be carried out automatically, periodically pausing for manual evaluation of designs by the human designer, and continued. This loop is repeated for p number of iterations, or until the designer is satisfied with the results of the search, and progresses to Step 12 by selecting a final design or final set of designs.

Chapter 4 explains Steps 1 through 7, including the *Numerical Optimization Search Strategy*, encompassed by the dashed line in Figure 3.1. Chapter 5 explains Steps 6 through 11, including the *Preference-guided Search*, which is encompassed by the dotted line in the same figure. Both of these chapters are also the main subjects of other works, [13] and [14], but are presented together in this thesis and explained in the context of the entire design methodology.

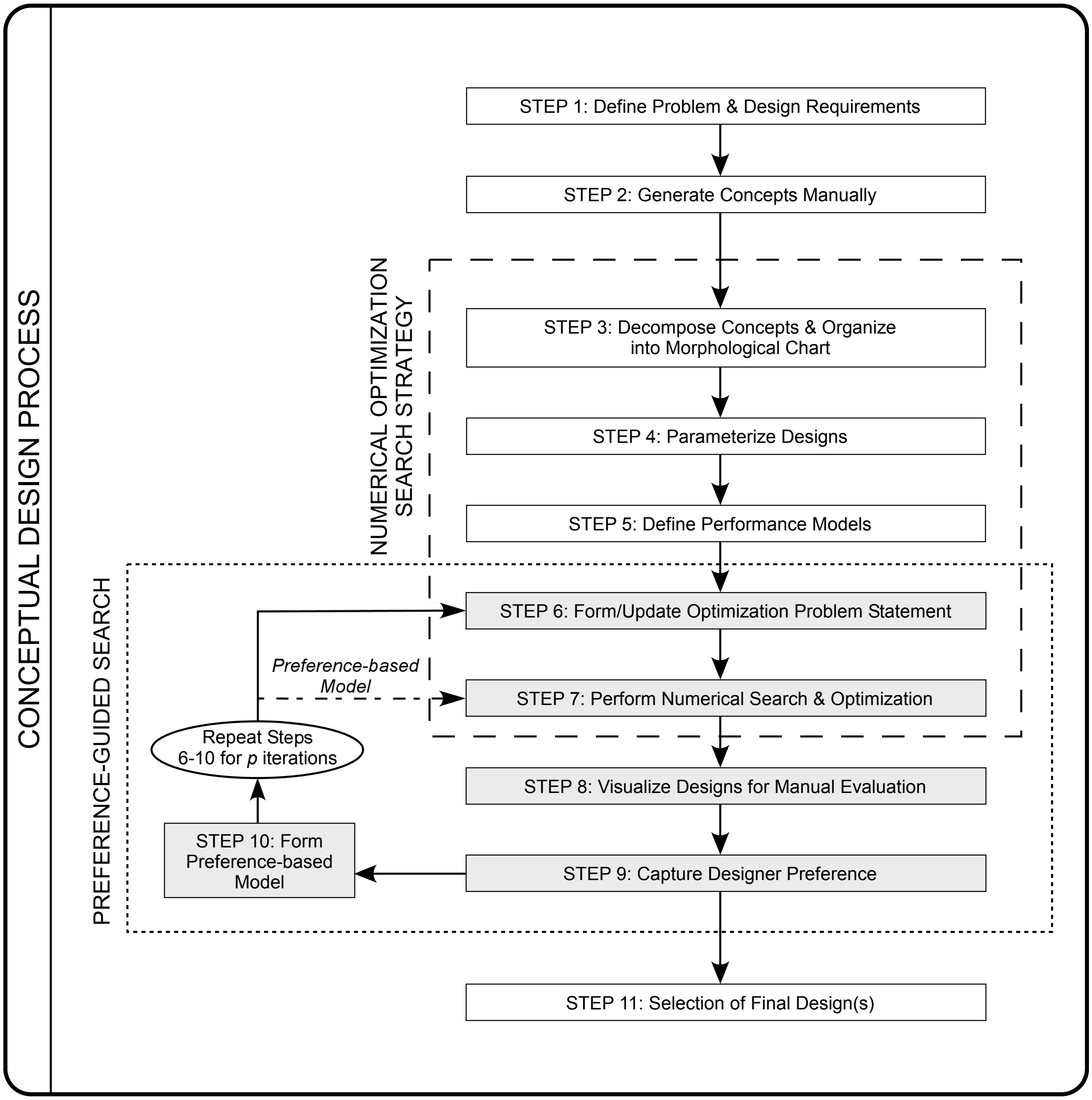


Figure 3.1: The conceptual design context within which the 11-Step Computationally-assisted Design Methodology fits.

CHAPTER 4. NUMERICAL OPTIMIZATION SEARCH STRATEGY

This chapter details Steps 1 through 7 of the computationally-assisted design methodology presented in this thesis. Steps 1 through 5 are intended to be completed one time by designers. Steps 3 through 7 are referred to as the Numerical Optimization Search Strategy, as shown in Figure 3.1. Steps 6 and 7 are completed automatically through computation, and will be repeated automatically as part of the Preference-guided Search process which will be explained in Chapter 5. The Numerical Optimization Search Strategy is now explained in Steps 1 through 7.

4.1 Step 1: Define Problem & Design Requirements

Defining the design problem and the design requirements – conditions that a concept must satisfy – involves understanding the customer needs and translating them into functional product specifications and design objectives $(\mu_1, \mu_2, ..., \mu_{n_\mu})$. Design requirements may be quantitative, requiring calculated performance levels, as well as qualitative in nature, using subjective judgment and intuition to evaluate performance. Traditional methods, such as customer surveys, lead users, and Quality Function Deployment, can help the designer discover the latent needs, create technical specifications, and create a product requirements list [1–4].

4.2 Step 2: Generate Concepts Manually

After defining the design requirements, designers generate concepts that are intended to meet one or more design requirements. Note that this step calls for *manual* generation and feature recombination methods, as opposed to *automatic* methods, which are described in Step 7. To manually generate concepts, the designer may use any effective method at his/her disposal – many such methods are discussed in the literature [1–4]. The initial pool of concepts generated can be comprehensive or focused. Concepts should, however, be sufficiently described to make it clear what the important features are and how they meet one or more design requirements. This

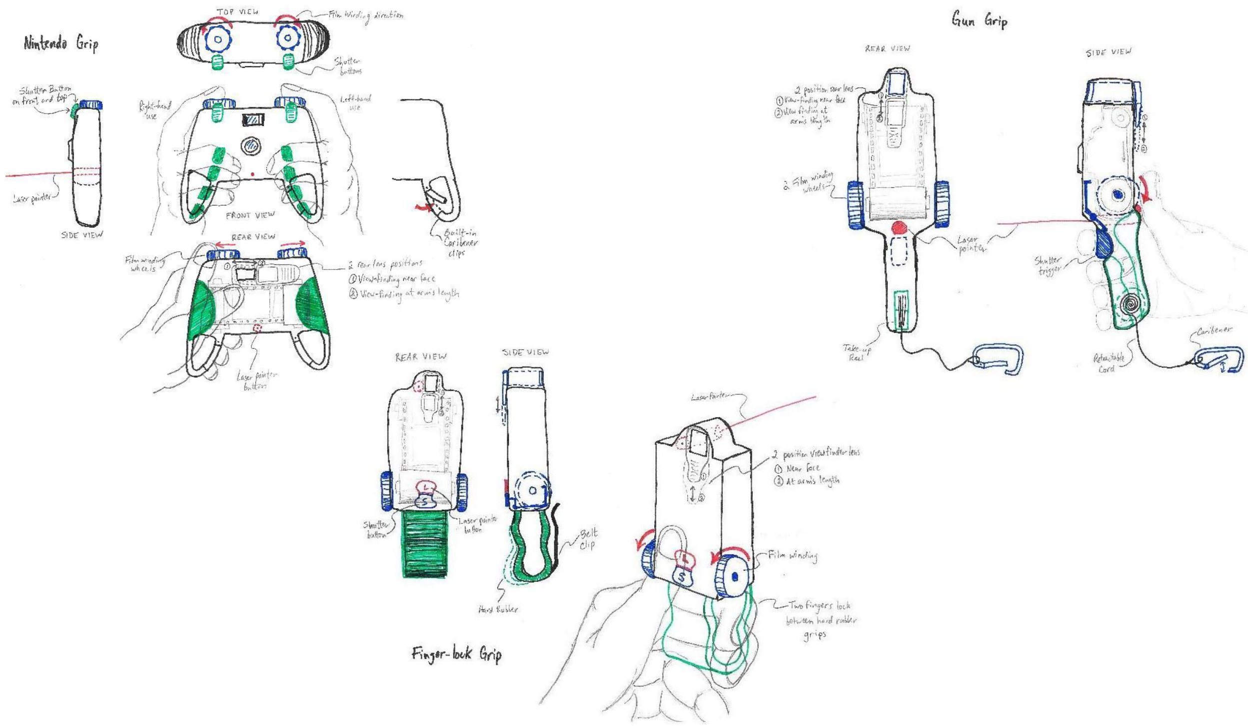


Figure 4.1: Manually generated concept sketches of a disposable camera product.

could be as simple as a descriptive list of concepts and their features, or as complex as detailed, dimensioned sketches with key features emphasized. Figure 4.1 shows sketches of three concepts that were manually generated by designers for a disposable camera product, including detailed notes describing the features and their intent to meet design requirements. While there are common features in this set of concepts (i.e. buttons, film advancement wheels, viewfinder) and several unique features on each concept (i.e. two handle grip, finger loop, tethered clip), it can be seen that each concept is unique because of the specific combination of features and parameters present in each one.

4.3 Step 3: Decompose Concepts & Organize into Morphological Chart

This step involves decomposing each initial concept into its basic features and subfunctions, and organizing them into a morphological chart. Chapter 2 reviews several decomposition strategies, any of which could be used. Here, the purpose of the decomposition in this step is to categorize concept features or attributes according to their intent so that interchangeable features

Table 4.1: Morphological chart example for a bicycle-like vehicle design problem.

Functions	**Possible Solutions**			
Power source	Human powered	Gas motor	Electric motor	Human-electric hybrid
Support rider	Small seat	Banana seat	Bucket seat	Suspended hammock
Contact ground	One wheel	Two wheels	Three wheels	Four wheels
Paint color	Red	Yellow	Blue	
...	...			

can be placed on the same row of a morphological chart and new concepts can be formed with combinations of features from the chart.

The decomposition method used by the new design methodology presented in this thesis, organizes the concept features based on function, form, or aesthetics. Form and aesthetics have been included to demonstrate the application of the new design methodology to early conceptual design work, when creative, non-numerical design features and attributes are present. The new design methodology is also applicable to design embodiment work, when features and parameter values are more refined. Generically, the features of the decomposed concepts can be organized into the morphological chart following the format previously shown in Fig. 2.1. Table 4.1 shows an example of how the decomposed functions of a bicycle-like vehicle could be organized into a morphological chart. It is also possible at this time, to include additional function solutions that were not present in the initial set of design concepts, but would be interchangeable with those already present.

It is well known that there is more than one way to decompose each concept [50]. No exception is found here. The list of concept features decomposed by one designer may be slightly different than another – resulting in a different morphological chart. At any time, designers may return to the decomposition step to redefine the design features as they recognize better ways to represent the intent of the initial set of concepts. The important point to remember is that the degree of detail to which designers decompose the concepts will be the degree of detail found in new designs that are formed automatically in Step 7 and on.

4.4 Step 4: Parameterize Designs

The features in the morphological chart produced in Step 3 and any other design variables now need to be *parameterized*, or represented numerically, to enable their use in the numerical search of the design methodology.

For computational purposes, the morphological chart is embodied in a two-dimensional matrix. Consequently, the feature descriptions are replaced with numerical values in a mathematical matrix. A generic example of a morphological chart put in matrix format is shown in Equation 4.1 as matrix F_m, along with its numerical equivalent, which will be referred to as a *morphological matrix*.

$$F_m = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1(n_{F_M})} \\ F_{21} & F_{22} & \cdots & F_{2(n_{F_M})} \\ \vdots & \vdots & \ddots & \vdots \\ F_{(n_R)1} & F_{(n_R)2} & \cdots & F_{(n_R)(n_{F_M})} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & \cdots & m_1 \\ 1 & 2 & 3 & \cdots & m_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \cdots & m_{n_R} \end{bmatrix} \tag{4.1}$$

The size of the matrix F_m is n_R by n_{F_M}, where n_R, is the number of function rows in the morphological chart, and n_{F_M}, is the maximum number of features in any of the rows.

Continuing with the bicycle-like vehicle example in the morphological chart in Table 4.1, Figure 4.2 shows how the information in the morphological chart transfers into a morphological matrix, with numbers representing the design features that were decomposed from the initial set of concepts.

In addition to the features in the morphological chart, a set of continuous variables may be needed to fully describe designs numerically. Equation 4.2 shows matrix F_c which contains the lower and upper bounds for each continuous variable.

$$F_c = \begin{bmatrix} F_{c1_l} & F_{c1_u} \\ F_{c2_l} & F_{c2_u} \\ \vdots & \vdots \end{bmatrix} \tag{4.2}$$

The method of parameterization presented herein uses a chromosome-like numerical representation of designs – a column vector of discrete values from each row of F_m and continuous values from within the bounds in each row of F_c – which is called the *design chromosome*. Gener-

Function	Possible Solutions			
Power source	Human powered	Gas motor	Electric motor	Human-electric hybrid
Support rider	Small seat	Banana seat	Bucket seat	Suspended hammock
Contact ground	One wheel	Two wheels	Three wheels	Four wheels
Paint Color	Red	Yellow	Blue	*
...	...			

$* =$ *empty index*

$$F_m = \begin{bmatrix} \text{Human powered} & \text{Gas motor} & \text{Electric motor} & \text{Human-electric hybrid} \\ \text{Small seat} & \text{Banana seat} & \text{Bucket seat} & \text{Suspended hammock} \\ \text{One wheel} & \text{Two wheels} & \text{Three wheels} & \text{Four wheels} \\ \text{Red} & \text{Yellow} & \text{Blue} & * \\ \dots & \dots & \dots & \dots \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & * \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

Figure 4.2: A morphological chart of features is parameterized into a numerical morphological matrix.

ically, the design chromosome is shown here

$$c = \begin{bmatrix} x_{d_1} \\ x_{d_2} \\ \vdots \\ x_{c_1} \\ x_{c_2} \\ \vdots \end{bmatrix} \tag{4.3}$$

The numerical values contained within the design chromosome which represent each discrete variable and continuous variable, will be defined as the *genes* of a design chromosome.

4.5 Step 5: Define Performance Models

The purpose of this step is to define the quantitative performance models, which can each be physics-based or preference-based, that will be used to automatically evaluate new designs that are found by the numerical search and optimization in Step 7. The output of the performance models should be a measure of how well designs meet the design requirements defined in Step 1. Each performance model μ is defined as

$$\mu = f(x_d, x_c) \tag{4.4}$$

For clarity, a *physics-based* model is defined as μ_n^{phys} within the set of all physics-based models used, as

$$\mu^{\text{phys}} = (\mu_1^{\text{phys}}, \mu_2^{\text{phys}}, ..., \mu_{n_{\mu_{\text{phys}}}}^{\text{phys}}) \tag{4.5}$$

and define a *preference-based model* as μ_m^{pref} within the set of all preference-based models used, as

$$\mu^{\text{pref}} = (\mu_1^{\text{pref}}, \mu_2^{\text{pref}}, ..., \mu_{m_{\mu_{\text{pref}}}}^{\text{pref}}) \tag{4.6}$$

Initially, the known performance models will likely include only the physics-based models that are applicable to the design problem at hand. If there is already a known quantitative preference-based model, it can be used immediately. However, this scenario will not be discussed further, but rather focus on the scenario of having an unmodeled preference-based design requirement and use the preference capture methods presented in Steps 9 through 10 to interactively form a quantitative preference-based model for use in the numerical search and optimization explained in Step 7. Some, or all of the performance models within the set $\mu = (\mu^{\text{phys}}, \mu^{\text{pref}})$ will become the design objectives in the optimization problem statement defined in Step 6.

The discrete nature of Morphological Charts

The discrete nature of the morphological chart can make it difficult to use one single performance model for all possible combinations of features. In fact, some combinations of features may form concepts that use very different physics-based models, parameters, and assumptions to calculate performance. This results in a more complex series of equations to calculate performance, but is permitted as long as any performance model that is used produces comparable performance values according to the common design objectives $(\mu_1, \mu_2, ..., \mu_{n_\mu})$. Using the example of the bicycle-like vehicle design problem, the physics-based performance of a gas motor and the electric motor are calculated using very different physics-based models and parameters, but can be compared because they both produce equivalent measures according to the physics-based objective to produce *power* for the vehicle.

4.6 Step 6: Form/Update Optimization Problem Statement

The purpose of this step is to form, or update, an optimization problem statement that will be used in Step 7 to search for the best performing designs. This step refers back to the generic multiobjective optimization problem statement introduced in Equation 2.5, which is subject to Equations 2.6, 2.7, and 2.8.

As mentioned in the previous section, in this work it is assumed that the physics-based models are known and it is desirable to form a quantitative preference-based model. To accomplish this, initially the physics-based models are left out of the optimization problem statement for a specified number of learning cycles. This *learning period* gives time for the numerical search to explore the design space and allow the designer to manually evaluate designs from the entire design space, unrestricted by the feasible design space defined by the physics-based models. During the learning period, the multiobjective optimization problem statement reduces to a single objective optimization problem statement, shown here:

$$\min_{x} \left\{ -\mu^{\text{pref}}(x_d, x_c) \right\} \tag{4.7}$$

When the learning period ends, a decision which will be discussed in Chapter 5, the optimization problem statement is updated to to be a multiobjective optimization problem statement, incorporating the physics-based models and the newly formed preference-based models, as shown here:

$$\min_{x} \left\{ -\mu^{\text{phys}}(x_d, x_c), -\mu^{\text{pref}}(x_d, x_c) \right\}. \tag{4.8}$$

subject to constraints listed in Equations 2.6, 2.7, and 2.8.

During the learning period, or after the learning period, the purpose of the optimization problem statement is to direct the numerical search and optimization in Step 7 to find the best performing designs according to those performance models included within the current form of the optimization problem statement. The creation of the preference-based model is explained in detail in Step 10, but here, it is sufficient to say that once it is formed, it is used in an equivalent manner as any of the physics-based models all contained in the complete set of design objectives $\mu = (\mu^{\text{phys}}, \mu^{\text{pref}})$.

4.7 Step 7: Perform Numerical Search and Optimization

Guided by the current optimization problem statement, a numerical search and optimization is now performed to find designs that will be presented to designers in Step 8. For reasons discussed at the end of Section 2.3, genetic algorithms are used as the numerical search method used in this thesis, although alternative search strategies and optimization methods could be used such as particle swarm [51, 52], other evolutionary methods [53, 54], or various others that can handle a mixed set of discrete and continuous variables.

The numerical search begins by creating an initial population of designs, in the form of concept chromosomes. On the initial iteration of the search, the population is created in a random manner. On subsequent iterations, a portion, or all, of the designs present at the end of the previous iteration can be used in this step. This replicates the inheritance principle of evolutionary algorithms, with the intention of carrying over the best design traits into successive iterations in order to continue to improve the performance of the designs generated by the methodology. When created randomly, the discrete genes in the design chromosome, such as those corresponding to discrete values from F_m, are randomly chosen from the integer values in each row of F_m. The genes that correspond to continuous variables are selected randomly from values within their allowable ranges.

Next, the physics-based models and/or preference-based models which are included in current optimization problem statement are used to evaluate the performance of the current population of designs. These performance values are used to calculate a *Maximin* fitness value for each design. The Maximin fitness function was introduced in Section 2.3, and is used because of the way it directs genetic algorithms to find a diverse set of pareto-optimal solutions [48]. The Maximin fitness value of each design will be what determines the probability that a design will be selected to be a parent design for reproduction.

After fitness values have been calculated for the entire population of designs, tournament selection is used to select parents for reproduction. A number of individuals equal to a specified tournament size are selected randomly from the population. Since the Maximin fitness function is a minimizing function the design with the lowest fitness value is selected to be a parent. This is repeated to select a second parent for reproduction. This is repeated until the number of parents is equal to the number of individuals in the current population.

Next, children designs are produced through a uniform, gene by gene crossover method for all discrete genes. Blend crossover [49] is used on genes that represent continuous variables, making it possible for children designs to receive random values anywhere in between the mother value and father value. Mutation is an operation that introduces a random change to each gene of a design in the chance that it will improve as a result of that change [49]. Mutation occurs to a gene if a randomly generated probability is less than a specified probability of mutation.

After reproduction has been completed, fitness values are recalculated considering the entire group of parents and children. The designs with the best fitness values are selected to form the next generation of designs. This form of *elitism* ensures that the best of both groups survive, helping the search to more quickly converge on the best designs. Again, because the Maximin fitness function is a minimizing function, the designs that have better (lower) fitness values have more probability of being selected as parents and more probability of continuing on to the next generation. Note, that this does not ensure that the children will always perform better than the parents, but does ensure that if designs improve they will have a better chance of survival than those that don't improve.

This genetic algorithm evolution optimization process repeats until a termination condition is met, which can be a specified number of generations or until a convergence criteria has been met, such as a minimum amount of change in objective values from one generation to the next. In the case of the Maximin fitness function, the progression of fitness scores from more negative to less negative indicates that the Pareto designs are becoming less clustered. In other words, the Pareto designs are more evenly spread out over the Pareto frontier.

Output of the Numerical Optimization Search Strategy

At this point, the numerical optimization search strategy identified in Steps 2 through 7 of Figure 3.1 is complete. A set of randomly created designs were used as starting points for the optimization to explore the design space and converge on a set of optimally performing designs. New combinations of features have been automatically formed and evaluated. Genetic algorithm optimization methods have evolved a set of designs to be optimally performing according to the performance models included in the current optimization problem statement, and produced a set of designs which is ready for manual, subjective evaluation by human designers. The capture and incorporation of designer preferences into the design methodology will now be explained in the next chapter.

CHAPTER 5. PREFERENCE-GUIDED SEARCH

This chapter details Steps 8 through 11 of the computationally-assisted design methodology, and is identified on Figure 3.1 as the *Preference-guided Search*. This is an iterative process, using the designs which were automatically formed, evaluated, and optimized in Steps 6 and 7 to capture designer preferences and incorporate those preferences into successive iterations of the same process. This phase is completed when a final design, or set of designs, is selected for further development. The Preference-guided Search in Steps 8 through 11 is now explained.

5.1 Step 8: Visualize Designs for Manual Evaluation

After the stopping criteria for the numerical search and optimization in Step 7 has been met, a subset of designs from the final generation/population, n_V number of designs, is presented to the human designers for subjective evaluation. The features present in this subset should representative of the population, allowing the designers to consider a diverse set of feature combinations. A smart-Pareto filter [39, 55] is one way to eliminate designs that are too similar to other designs based on the relative closeness of their objective values. It may also be advantageous to present the set of designs with the best performance according to the preference-based models, or any of the physics-based models.

In addition to the designs selected through a filtering strategy, a percentage of the designs presented to the designer, r_R, should be randomly selected. This helps maintain a level of diversity within the set of designs presented, especially during the learning period as the population of designs gradually converges on similar designs that match the designers preference.

To help designers quickly comprehend the make-up of the designs being evaluated, the calculated performance values of the physics-based models and preference-based models are presented visually along side graphical representations of designs. The graphical representations can be created through CAD or other parametric software that can quickly generate the visual images.

Having the performance data and the images of the designs shown together helps designers make trade-offs between the quantitative performance and qualitative aspects of the designs as they select their preferred designs.

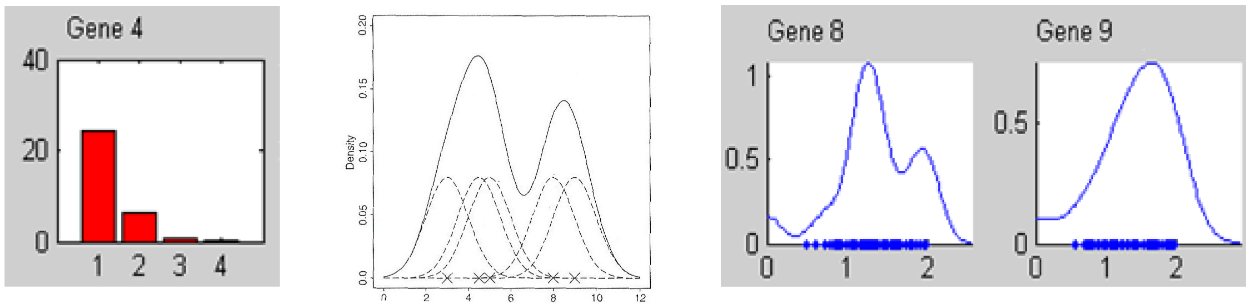
5.2 Step 9: Capture Designer Preferences

With the visual representations of the designs presented along with the performance levels, designers now select the designs that he/she prefers. One purpose for having human designers manually evaluate designs is to attempt to capture any unmodeled objectives. Also, humans can very quickly make mental trade-offs of competing objectives, resulting in subjective decisions. Evaluation methods such as rating, ranking, or scoring of the designs could be used to indicate preference. However, the minimum level of rating is to *select* or *not select* individual designs as preferred, which is the rating/evaluation method used in this work. In each successive round of manual evaluation, the design chromosomes of the designs which are selected as preferred are recorded. The feature frequency count and variable values for each preferred design will be used next, in Step 10, to form a quantitative preference-based model.

5.3 Step 10: Form Preference-based Model

In order to automatically evaluate a designer's preference for designs which were formed in Step 7, a mathematical model is now formed to predict the designer's preference for certain features and parameter values. In order to quickly form a preference model, statistical probability is used as the underlying theory to predict the probability that a design will be preferred by the designer. These statistical methods were introduced in Section 2.3. For both discrete and continuous genes in the design chromosome, the gathered preference data is used to estimate the probability density functions.

When creating an individual preference model for the discrete genes, a *discrete probability density estimate,* also called the probability mass estimate, is created. Figure 5.1 (a) shows an example of a histogram of the number of times that each value was present in the preferred designs from Step 9. The histogram is actually a graphical estimate of the real probability density function.



(a) (b) (c)

Figure 5.1: Examples of (a) an estimated probability mass function for a discrete variable and (b) an estimated probability density function for a continuous variable, used to model and predict a designer's preference for the variables.

The probability estimate $\hat{f}$ for a specific gene value is calculated using Equation 2.2, which is equal to the proportion of previously recorded genes with the same value.

When creating an individual preference model for the continuous genes, density estimation and smoothing techniques [32, 34] are used, which empirically form a distribution through the summation of normal distributions around each data point. The gene values present in the preferred designs from Step 9 are used in Equation 2.4 to form the probability density estimate. Figure 5.1 (b) shows an example of a probability density estimate for a continuous variable and the points used to create it. This empirical approach to creating the preference function, allows the model to be updated each time evaluations are completed by the designer, thus continually improving the accuracy of the model.

The individual preference models for each gene in the design chromosome can be combined into a single preference model used to predict preference for an entire design, as defined here:

$$\mu_m^{\text{pref}} = \left(\hat{f}_{x_{d_1}}(x_{d_1}) \cdot \ldots \cdot \hat{f}_{x_{d_i}}(x_{d_i}) \cdot \hat{f}_{x_{c_1}}(x_{c_1}) \cdot \ldots \cdot \hat{f}_{x_{c_j}}(x_{c_j}) \right) \tag{5.1}$$

where $\hat{f}_{x_{d_i}}$ is the probability density estimate for the i-th discrete variable/gene in a design chromosome, and $\hat{f}_{x_{c_j}}$ is the probability density estimate for the j-th continuous variable/gene in a design chromosome. This combined preference model μ_m^{pref}, after being sufficiently developed, can be used as a preference-based objective in the optimization problem statement when it is updated in

Step 6, to guide the numerical search towards designs with a higher probability of being preferred by designers.

5.4 Step 11: Selection of Final Design(s)

As with a conventional conceptual design process (See Figure 1.1 (a)), the use of the computationally-assisted design methodology in the conceptual design process (See Figure 1.1 (b)) produces multiple iterations of designs until the designer is satisfied with the results. Each iteration of the Preference-guided Search in Steps 6 through 10 gathers more data with which it can update the preference-based model, gradually improving the ability of the numerical search to find designs that will be preferred by the designer. It is generally accepted in the field of Interactive Evolutionary Computation (IEC) research that when subjective human preference is involved there is not a global optimum represented by a single design [31]. For this reason, the new computationally-assisted design methodology attempts to thoroughly search for a set of designs in a global optimum area. When a designer is satisfied with the final set of designs, any number of designs from that set can be used as a starting point for the next phase of the product development process or as a spark for further conceptual design efforts.

Stopping Criteria for Iterations

During the learning period, the preference-based model will be updated and improved with each iteration of Steps 6 through 10, and the search will gradually converge on a set of similar designs that match the preference of the designer. The decision to end the learning period can be made by the designer when he/she feels that the individual gene preference models have been accurately captured, or a stopping criteria has been met, such as when all of the designs selected for visualization (See Step 8) which are not random selected, $n_F = n_V(1 - r_R)$, have preference-based performance scores above a designer specified amount, such as 60%. In iterations after the learning period, a stopping criteria could be determined in a similar fashion as an alternative to stopping at the discretion of the designer.

CHAPTER 6. PRODUCT EXAMPLES

In this chapter, the computationally-assisted design methodology developed in the previous chapters is applied to two product design scenarios to demonstrate the ability to accomplish the objectives of this thesis, as outlined in Chapter 1. The first example is the design of a table, which has very basic functional requirements, but also has a large aesthetic component to its design. The table example is demonstrated on Steps 1 through 8 of the design methodology. The second example is the design of a vehicle platform, which also has functional and aesthetic components to the design. The vehicle platform example is demonstrated on the entire methodology, Steps 1 through 11. This includes the use of the Preference-guided Search, demonstrating the ability of the design methodology to handle more complex engineering systems that are very time consuming and challenging to optimize through manual processes.

6.1 Example 1: Table Design

Consider the design of a table with a single work surface at a fixed height. The table must be free standing and be made of common table materials. The main customer need and functional purpose of the table is to provide a usable working surface in a small room. The table must be stable to work on and should not cost too much.

Example 1: Step 1

The functional design requirements of the table are captured in two physics-based performance models: (1) surface area of the table top and (2) the cost of materials. The qualitative requirement for the table could be captured in a preference-based model to evaluate (3) aesthetic appeal. However, in this section, only the physics-based models will be used in the numerical search and optimization. The preference capture and incorporation methods of the Preference-guided Search will be demonstrated in the next example. Table 6.1 shows these design objectives

Table 6.1: Design Objectives for table concept example.

	Objective	units	direction	range
1	Surface Area, μ_1	cm^2	Maximize	$0 < \mu_1 \leq w_R \cdot l_R$
2	Cost, μ_2	\$	Minimize	$0 < \mu_2$
3	Aesthetics, μ_3	n/a	Maximize	n/a

Table 6.2: Design parameters for table design example.

Parameter	Possible Values
Top Thickness, t_T	1 - 10 cm
span1, w_1	0 - 300 cm
span2, w_2	0 - 400 cm
leg diam/width, d	1 - 20 cm
base radius, r_B	0 - 200 cm
base thickness, t_B	1 - 20 cm
base width, w_B	1 - 200 cm

and their units of measure. Additionally, other design parameters and constants needed in the performance calculations for the table are shown in Table 6.2 and Table 6.3, respectively.

Example 1: Step 2

Figure 6.1 shows several sketches of manually generated table concepts, with notes indicating distinguishing features and details such as the material of the top, and the shape of the legs.

Example 1: Step 3

The manually generated concepts are decomposed by features of function and form, and then organized into the morphological chart, shown in Table 6.4.

Table 6.3: Fixed parameters for table concept example.

Constant	**Value**	**units**
Room width, w_R	300	cm
Room length, l_R	400	cm
Wood price, C_w	0.002745	\$/cm^3
Acrylic price, C_w	0.016000	\$/cm^3
Glass price, C_w	0.015000	\$/cm^3
Steel price, C_w	0.024000	\$/cm^3
Wood density, ρ_w	0.000750	kg/cm^3
Acrylic density, ρ_a	0.001190	kg/cm^3
Glass density, ρ_g	0.002600	kg/cm^3
Steel density, ρ_s	0.007850	kg/cm^3
height of table, h_T	100	cm

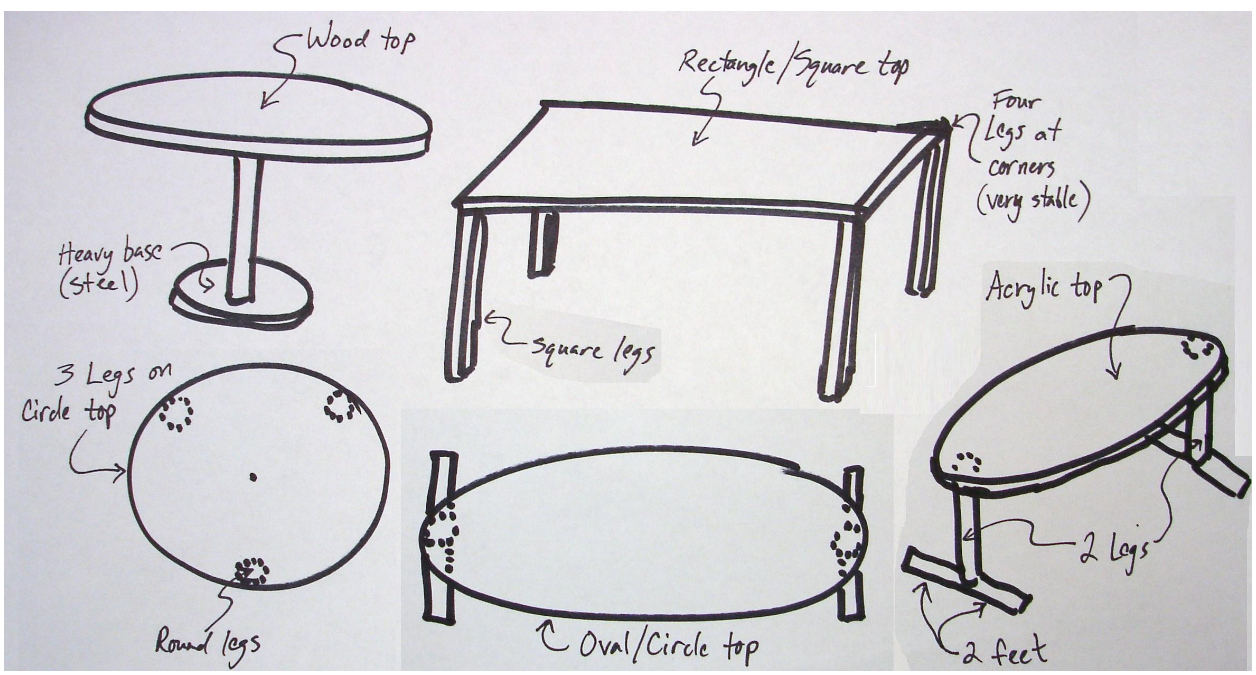


Figure 6.1: Sketches of manually generated table concepts.

Table 6.4: Morphological chart for table concept example.

Feature	**Possible Solutions**					
Top Shape	circle	oval	square	rectangle		
Top Material	Wood	Acrylic	Steel	Glass		
Leg Style	square leg	round leg				
Leg Quantity	1	2	3	4		
Leg & Base Material	Wood	Acrylic	Steel			
Base Style	circular base	2 feet	3 feet	4 feet	square base	none

Example 1: Step 4

To parameterize the designs for use in the numerical search, first, the numerical form of the morphological chart is created, which is shown in Equation 6.1 as matrix F_m.

$$F_m = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} & & \\ F_{21} & F_{22} & F_{23} & F_{24} & & \\ F_{31} & F_{32} & & & & \\ F_{41} & F_{42} & F_{43} & F_{44} & & \\ F_{51} & F_{52} & F_{53} & & & \\ F_{61} & F_{62} & F_{63} & F_{64} & F_{65} & F_{66} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & & \\ 1 & 2 & 3 & 4 & & \\ 1 & 2 & & & & \\ 1 & 2 & 3 & 4 & & \\ 1 & 2 & 3 & & & \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix} \tag{6.1}$$

Next, the set of variables x from the morphological matrix in Equation 6.1, and the set of parameters p from Table 6.2, are combined to form the design chromosome c, following the generic design chromosome set forth in Equation 4.3. The complete design chromosome for the table is defined as

$$c_{\text{table}} = \begin{bmatrix} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & t_T & w_1 & w_2 & d & r_B & t_B & w_B \end{bmatrix}^{\mathrm{T}} \tag{6.2}$$

Table 6.5 shows the complete list of the variables and parameters that make up the genes of the design chromosome. Also shown are the possible values for the genes, and an example design chromosome.

Table 6.5: A list of variables and parameters that make up the genes of the design chromosome for the table design example

Gene	Possible Values	Example Chromosome
Top Shape, F_1	1, 2, 3, 4	1
Top Material, F_2	1, 2, 3, 4	2
Leg Style, F_3	1, 2	1
Leg Quantity, F_4	1, 2, 3, 4	4
Leg & Base Material, F_5	1, 2, 3	2
Base Style, F_6	1, 2, 3, 4, 5, 6	4
Top Thickness, t_T	1 - 10 cm	0.050
span1, w_1	0 - 300 cm	1.1
span2, w_2	0 - 400 cm	2.0
leg diam/width, d	1 - 20 cm	0.110
base radius, r_B	0 - 200 cm	0.2
base thickness, t_B	1 - 20 cm	0.010
base width, w_B	1 - 200 cm	1.900

Example 1: Step 5

The two physics-based performance models, table surface area (μ_1), and material cost (μ_2), along with the unmodeled preference-based performance model for aesthetics (μ_3), are defined as

$$\mu_1 = A_t \tag{6.3}$$

$$\mu_2 = \sum_{p=1}^{n_p} C_p \tag{6.4}$$

$$\mu_3 = f_{\text{aesthetic}}(x_d, x_c) \tag{6.5}$$

where x_d is a set of discrete variables/genes and x_c is a set of continuous variables/genes to describe a design.

Example 1: Step 6

The single objective optimization problem statement used during the learning period of the Preference-guided Search uses only the preference-based model, μ_3, equivalent to Equation 4.7. The multiobjective optimization problem statement used after the learning period uses both physics-based models and the newly formed preference-based model, resulting in the following form of Equation 4.8, as shown here

$$\min_{x} \{-\mu_1(x_d, x_c), \mu_2(x_d, x_c), -\mu_3(x_d, x_c)\}. \tag{6.6}$$

subject to:

$$0 < \mu_1 \leq w_R \cdot l_R \tag{6.7}$$

$$0 < \mu_2 \tag{6.8}$$

subject to:

$$F_{\min} < \frac{1}{d_0} \sum_{p=1}^{n_p} M_p < F_{\max} \tag{6.9}$$

where w_R is the usable width of the room, l_R is the length of the room, $F_{\min}$ is the minimum downward tipping force, $F_{\max}$ is the maximum downward tipping force, d_0 is the distance from the outer edge of the table top to the tipping point on the closest point of the feet or legs, M_p is the moment of the p-th table part, A_t is the area of the table work surface, and C_p is the cost of the p-th table part. Figure 6.2 shows additional details of the model used to calculate the stability constraint for two generalized cases of table design.

Example 1: Step 7

The genetic algorithm used as the numerical search method used a generation size of $N = 50$, a crossover probability of $p_{\text{crossover}} = 0.6$, a mutation probability of $p_{\text{mutation}} = 0.01$, a tournament ratio of $r_{\text{tournament}} = 0.1$, and number of generations $G = 100$. These conditions for the numerical search result in the automatic evaluations of 5,000 designs/combinations of features and parameters.

Figure 6.3 shows the progression of the numerical search by plotting the average fitness score of each generation. A negative maximin fitness represents a non-dominated design, while a positive fitness represents a dominated design. Note that the average maximin fitness gradually approaches zero; when the maximin fitness approaches zero the designs in the population are more evenly distributed over the objective space. Figure 6.4 contains several representative generations

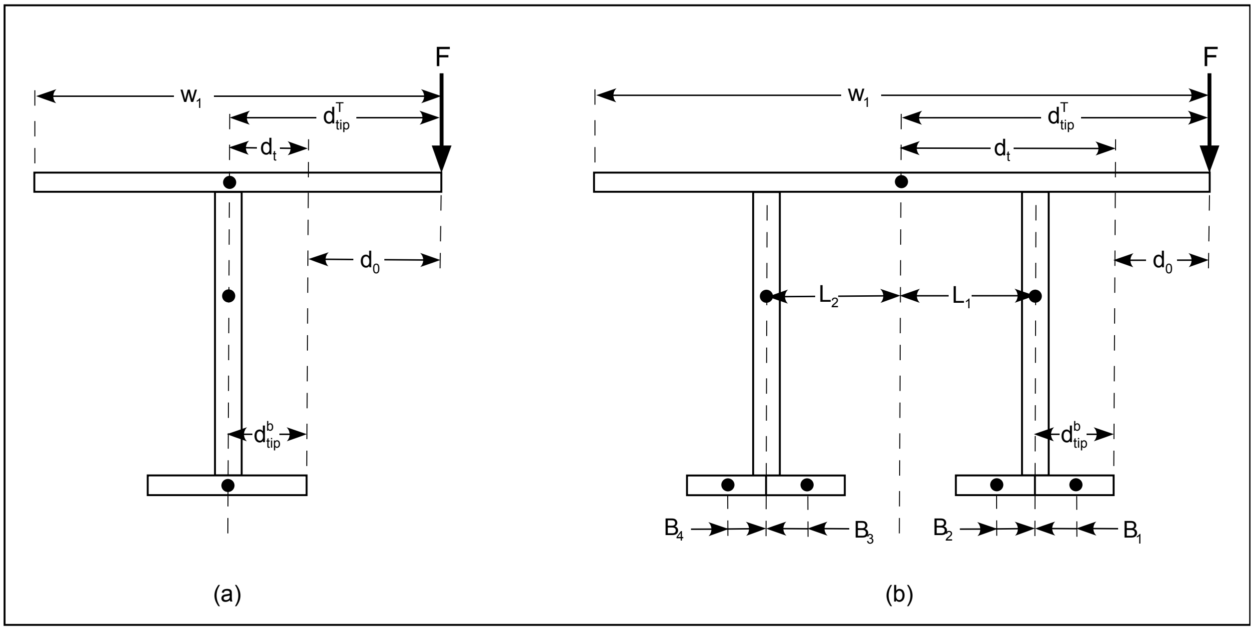


Figure 6.2: Dimensions used to calculate the stability for table designs with (a) one leg and (b) two legs

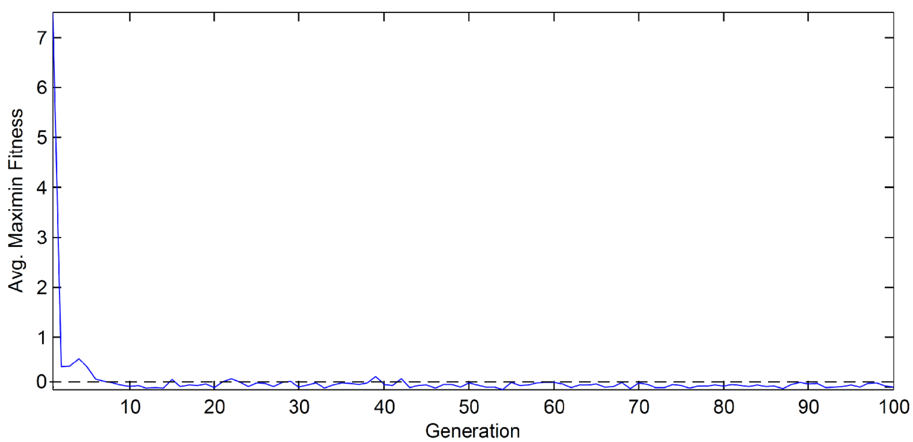


Figure 6.3: The progression of the average fitness value of each generation.

of designs, showing improving physics-based performance through the generations. Therefore, it can be seen that the numerical search and genetic algorithm optimization has found a population that is non-dominated and well distributed. The search is carried out over 100 generations to show convergence when the fitness score and objective values have little significant change.

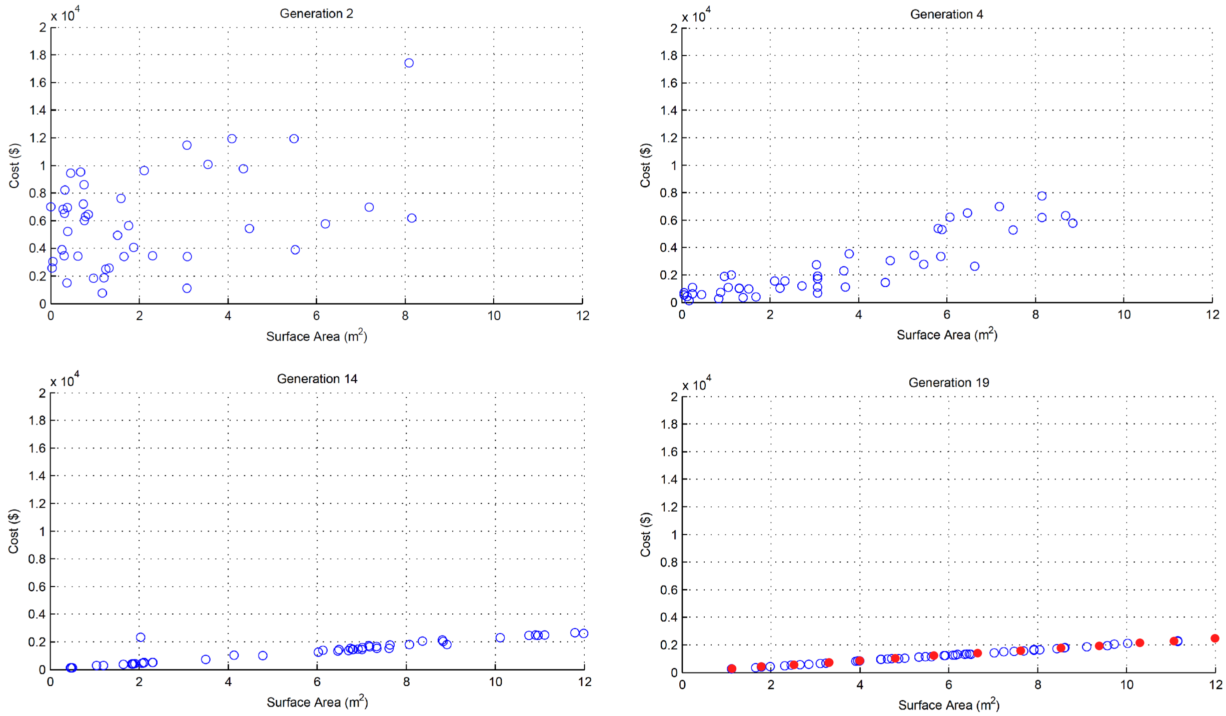


Figure 6.4: The progression of the performance of the designs through the generations.

After the genetic algorithm has completed its search, the concepts in the final generation are ready to present to the designer. The lower right plot in Figure 6.4 shows the designs of the final generation plotted in the two dimensions of the physics-based performance objectives. It should be noted that all 50 individuals of the final generation are *Pareto-optimal* designs. The designs represented by filled circles are the smart Pareto filtered designs that will be presented to the designers for manual, subjective evaluation.

Example 1: Step 8

Figure 6.5 shows visual representations of several table designs that were automatically formed. In these samples it can be seen that the table features are recognizable from the initial concept sketches (See Figure 6.1), however some of the combinations of features are new and were not previously considered by designers. After designers manually evaluate these results, their preferences will be captured and used to guide the numerical search and optimization in Step 7 of subsequent iterations of the Preference-guided Search.

Figure 6.5: Table designs automatically formed consisting of new combinations of table features.

Table 6.6 shows numerical representations of three sample designs from the final generation, showing their physics-based performance values and design chromosome values.

Example 1: Results

An initial set of creatively produced table concepts was decomposed and organized into a morphological chart. The numerical representation of the features in the morphological chart was successfully used in the numerical optimization search strategy to quickly search through 5,000 feature and parameter combinations to find new combinations to form new design concepts, all of which satisfy Pareto optimality conditions. Genetic algorithms were used to numerically search and form populations of new table designs that evolved toward better performing and more diverse designs. This is shown in Figure 6.3 by the trend of the average maximin fitness approaching zero.

Table 6.6: Three representative designs of the final generation, showing their physics-based performance values and design chromosome values.

Objective	**Design 1**	**Design 2**	**Design 3**	**units**
Surface Area, μ_1	8.7651	8.3501	0.7699	m^2
Cost, μ_2	204.3240	156.9177	87.867	$
Gene				
Top Shape, F_1	rectangle	rectangle	oval	
Top Material, F_2	Wood	Wood	Wood	
Leg Style, F_3	square leg	round leg	round leg	
Leg Quantity, F_4	1	4	4	
Leg & Base Material, F_5	Wood	Wood	Wood	
Base Style, F_6	none	none	none	
Top Thickness, t_T	0.84	0.68	0.57	cm
span1, w_1	221.35	221.51	28.41	cm
span2, w_2	395.98	387.46	345.06	cm
leg diam/width, d	2.97	1.56	1.87	cm
base radius, r_B	26.86	65.03	9.28	cm
base thickness, t_B	14.71	14.71	6.11	cm
base width, w_B	175.68	175.68	80.50	cm

The final population of designs is the result of one pass through the Numerical Optimization Search Strategy, identified in Steps 1 through 8 in the design methodology. The quantitative performance values and visual representations of the new designs are presented to designers for manual evaluation. All the designs presented are *Pareto-optimal* designs according to the physics-based performance objectives, and now, human designers can make preference-based trade offs in their evaluation of the designs.

6.2 Example 2: Vehicle Platform Design

Periodically, automobile manufacturers will produce a new type of vehicle that is considered innovative, by changing the form and/or function to a new look or use. Several modern examples like this are vehicles types such as the mini-van, the station wagon, the sport utility vehicle, and the crossover. Realistically, these vehicles are simply different combinations of existing features and parameters of other vehicles, but there was a market need for those type of new vehicles regardless if the need was based on functionality, aesthetic appeal, or some combi-

Table 6.7: Design Objectives for vehicle design example.

	Objective	units	direction	range
1	Price, μ_1	\$	Minimize	$0 < \mu_1$
2	Weight, μ_2	lbs	Minimize	$0 < \mu_2$
3	Seating, μ_3	n/a	Maximize	$0 < \mu_3$
4	Towing, μ_4	lbs	Maximize	$0 < \mu_4$
5	Cargo Space, μ_5	ft^3	Maximize	$0 < \mu_5$
6	Preference, μ_6	n/a	Maximize	$0 < \mu_6 < 1$

nation. While outwardly fairly simple, automobiles are very complex systems on the inside, using multiple, high-tech, integrated mechanical and electrical systems to produce a vehicle that functionally performs as specified. In this section the computationally-assisted design methodology is applied to a vehicle platform design example, and will demonstrate the ability of the methodology to capture designer preference for features and parameters of vehicle design, form and incorporate quantitative preference-based models with physics-based performance models into the search for, and evaluation of, significantly more designs and novel designs than could be done by manual methods.

Example 2: Step 1

A list of the vehicle platform design requirements and their boundary conditions is shown in Table 6.7. The requirements for this example are common criteria used when evaluating a vehicle to purchase. For reference, Figure 6.6 shows an image of a vehicle filtering tool with very similar vehicle selection criteria on the Ford Motor Company web site [56].

Example 2: Step 2

Figure 6.7 shows examples of the human-generated concept sketches for current vehicle types such as compact car, mid-size sedan, pick-up truck, passenger vans, and so forth.

Figure 6.6: A vehicle filtering tool on the Ford Motor Company web site [56]

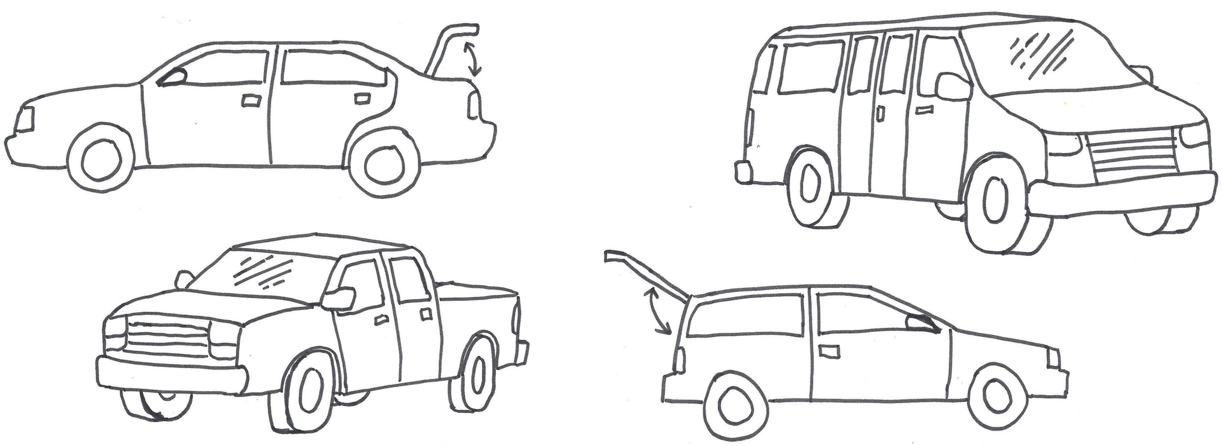


Figure 6.7: Human-generated concept sketches of vehicles.

Example 2: Step 3

The functions of these vehicle concepts, along with other common vehicle features and components, have been decomposed into the morphological chart, shown in Table 6.8. Additional design parameters are shown in Table 6.9, and design constants are shown in Table 6.10. The design features and other design parameters can be seen in in Figure 6.8, which contains a schematic of the vehicle platform design.

Table 6.8: Morphological chart for vehicle platform design example.

Feature	**Possible Solutions**					
Doors, F_1	2 doors	4 doors				
Chassis, F_2	compact	mid-size	full-size	heavy duty	super duty	
Engine, F_3	4-cylinder	V6	V8	V8 Diesel	electric	hybrid
Drive type, F_4	FWD	RWD	AWD	4WD		
Cargo style, F_5	rear hatch	truck bed	trunk			

Table 6.9: Design parameters for vehicle platform design example.

Parameter	**Possible Values**
Wheel base, w_B	72 - 175 in
nose length, n_L	25 - 50 in
tire diameter, t_D	20 - 36 in
cab height, s_{H2}	36 - 120 in
# of seat rows, n_r	1,2,3,4
tire aspect ratio, t_r	0.20 - 0.85
ground clearance, g_C	6 - 20 in

Table 6.10: Design constants for vehicle platform design example.

Constant	**Value**	**units**
2-seat width threshold, w_T	79	in
seat depth, d_S	40	in
material cost, C_m	0.002745	\$/cm^3

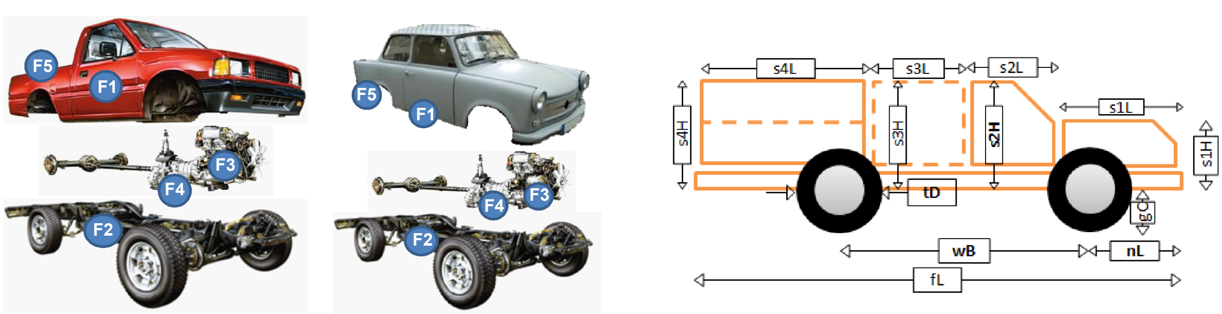


Figure 6.8: A schematic of the vehicle platform design example

Example 2: Step 4

The numerical form of the morphological chart is shown in Equation 6.10 as the morphological matrix, F_m.

$$F_m = \begin{bmatrix} F_{11} & F_{12} & & & & \\ F_{21} & F_{22} & F_{23} & F_{24} & F_{25} & \\ F_{31} & F_{32} & F_{33} & F_{34} & F_{35} & F_{36} \\ F_{41} & F_{42} & F_{43} & F_{44} & & \\ F_{51} & F_{52} & F_{53} & & & \end{bmatrix} = \begin{bmatrix} 1 & 2 & & & & \\ 1 & 2 & 3 & 4 & 5 & \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & & \\ 1 & 2 & 3 & & & \end{bmatrix} \tag{6.10}$$

The complete design chromosome is defined as

$$c_{\text{table}} = \begin{bmatrix} F_1 & F_2 & F_3 & F_4 & F_5 & w_B & n_L & t_D & s_{H2} & n_r & t_r & g_C \end{bmatrix}^{\text{T}} \tag{6.11}$$

Example 2: Step 5

The five physics-based performance models, Price (μ_1), Weight (μ_2), Seating (μ_3), Towing Capacity (μ_4), and Cargo Space (μ_5), along with the unmodeled preference-based performance model for aesthetics (μ_6), are defined as

$$\mu_1 = m_B \cdot C_m + C_t + C_{F2} + C_{F3}; \tag{6.12}$$

$$\mu_2 = m_B + m_t + m_{F2} + m_{F3} \tag{6.13}$$

$$\mu_3 = 3 \cdot n_r \tag{6.14}$$

$$\mu_4 = 1000 \cdot (F_2 - 1) \cdot \sqrt{F_2 (F_3 - 1)} \tag{6.15}$$

$$\mu_5 = c_r \cdot V_c \tag{6.16}$$

$$\mu_6 = f_{\text{aesthetic}}(x_d, x_c) \tag{6.17}$$

where x_d is a set of discrete variables/genes and x_c is a set of continuous variables/genes to describe a design.

Example 2: Step 6

The single objective optimization problem statement used during the learning period of the Preference-guided Search uses only the preference-based model, μ_6, equivalent to Equation 4.7. The multiobjective optimization problem statement used after the learning period uses both physics-based models and the newly formed preference-based model, resulting in the following form of Equation 4.8, as shown here

$$\min_x \{\mu_1(x_d,x_c),\mu_2(x_d,x_c),-\mu_3(x_d,x_c),-\mu_4(x_d,x_c),-\mu_5(x_d,x_c),-\mu_6(x_d,x_c),\}. \tag{6.18}$$

subject to:

$$0 < \mu_1 \leq 100,000 \tag{6.19}$$

$$0 < \mu_2 \leq 50,000 \tag{6.20}$$

$$2 \leq \mu_3 \leq 15 \tag{6.21}$$

$$0 < \mu_4 \leq 26,000 \tag{6.22}$$

$$20 \leq \mu_5 \leq 200 \tag{6.23}$$

where x_d is a set of discrete variables/genes and x_c is a set of continuous variables/genes to describe a design.

Example 2: Step 7

The genetic algorithm used as the numerical search method used a generation size of $N = 540$, a crossover probability of $p_{\text{crossover}} = 0.2$, a mutation probability of $p_{\text{mutation}} = 0.0001$, a tournament ratio of $r_{\text{tournament}} = 0.2$, and number of generations $G = 20$. These conditions for the numerical search result in the automatic evaluations of 10,800 designs/combinations of features and parameters, per execution of the numerical search in each iteration of the design methodology.

Figure 6.9 shows the progression and convergence of the objectives in the numerical search over 10 iterations. The first 3 iterations were the learning period, when only the preference-based

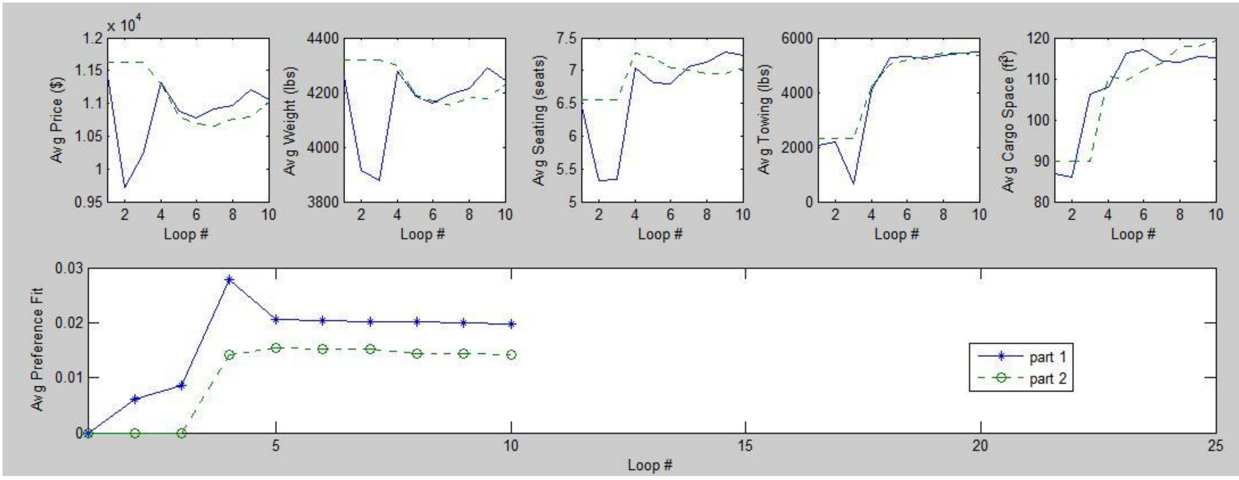


Figure 6.9: The progression of the design objectives through the Preference-guided Search, changing from single objective optimization to multiobjective optimization after iteration 3.

objective was used. After the preference-based model was developed, the physics-based objectives were included in the numerical search for designs, changing to a multiobjective search and causing trade-offs to be made between the competing objectives. After 10 iterations, all the designs were non-dominated, Pareto-optimal designs, and the objectives had converged, having found a set of designs that best met the objectives.

Example 2: Step 8

Figure 6.10 shows visual representations of a set of vehicle designs that were automatically formed, through random selection methods during the learning period of the design methodology when no optimization has taken place. The visual representations of the vehicles also presents each design's physics-based performance, preference-based performance, and several critical features descriptions, such as engine, chassis, and rear cargo style.

Example 2: Steps 9 & 10

After designers subjectively evaluate the designs which are displayed, the features and parameters present in the preferred designs are captured and used to form the preference-based models, using the methods explained in Chapter 5. An example of the preference models formed is shown in Figure 6.11.

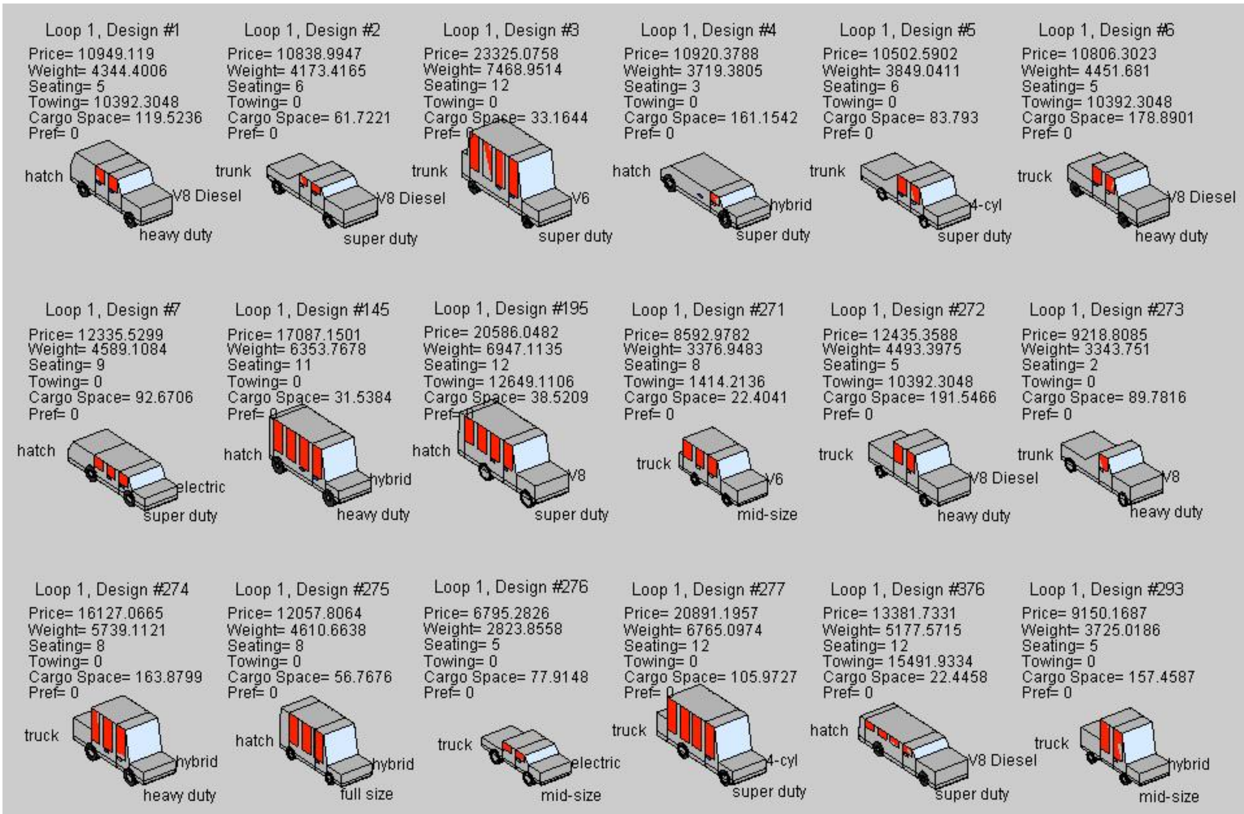


Figure 6.10: Non-optimized vehicle designs automatically formed and presented to human designers for subjective evaluation.

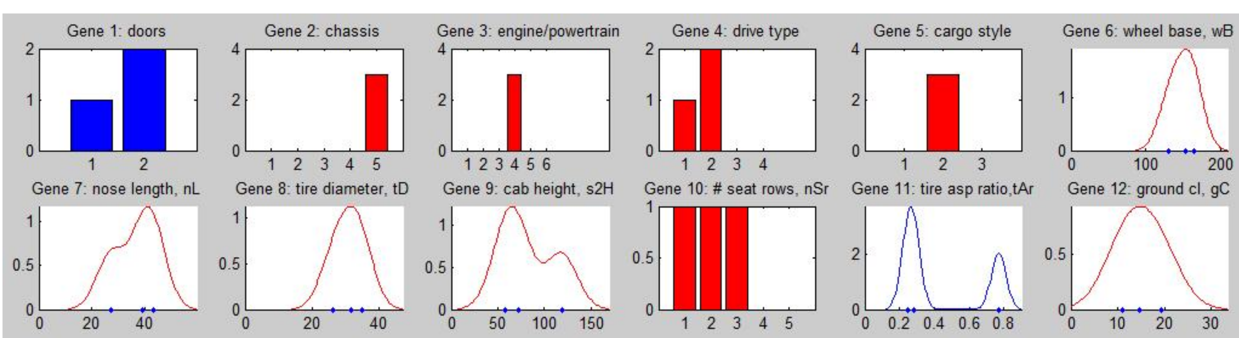


Figure 6.11: The preference-based models for each design gene, formed from subjective evaluation of a human designer.

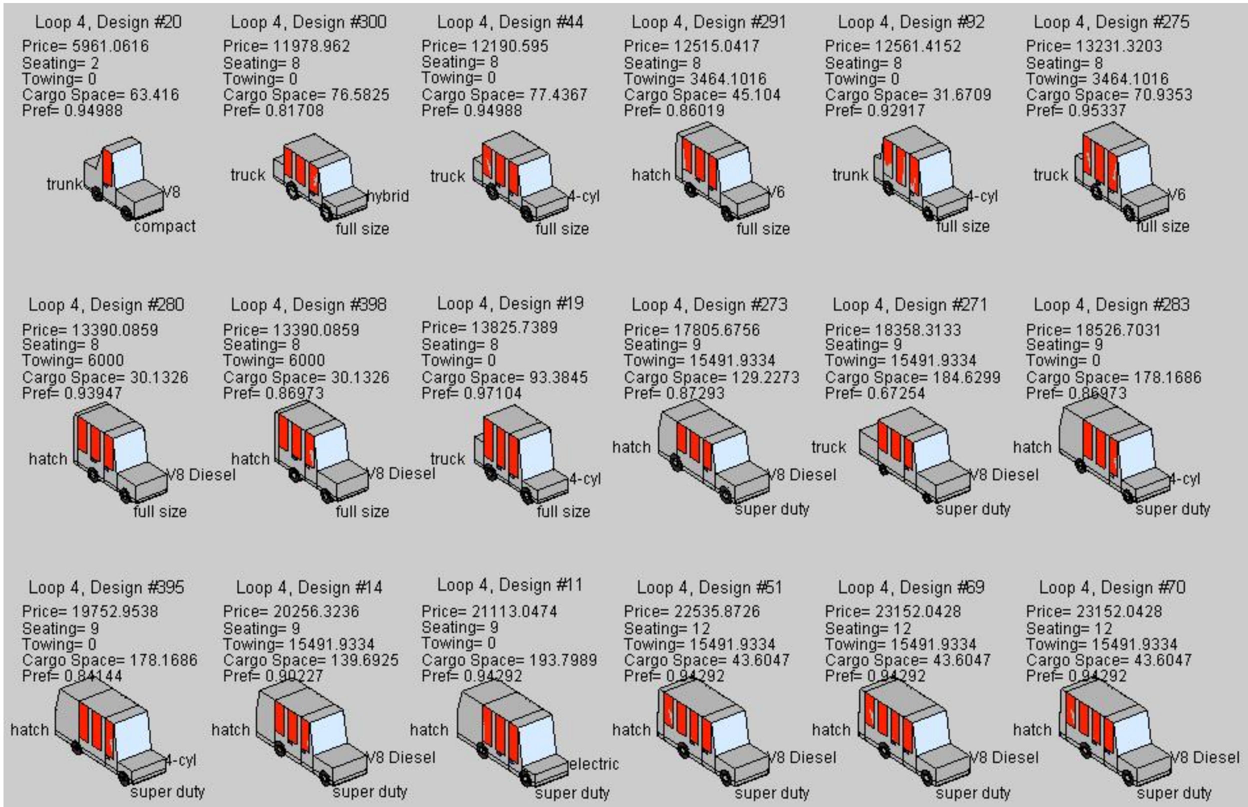


Figure 6.12: A set of vehicle designs that has converged using the preference-based models and physics-based models of performance.

Example 2: Step 11

After the Preference-guided Search has used the new preference-based model along with the physics-based models, the set of designs formed will begin to converge on designs with commonalities in some areas, but differences in others. Figure 6.12 shows a set of designs that is generally similar in appearance, but has different trade-offs of physics-based performance.

Example 2: Results

For quantitative validation purposes, an automated test was performed to measure the ability of the numerical search methods of the new design methodology to find more preferred designs. The test ran through the steps of the methodology, automatically selecting designs in Step 9 according to a certain criteria in order to simulate the choices that a human designer would make. The upper plot in Figure 6.13 shows an improvement in the average preference-based performance

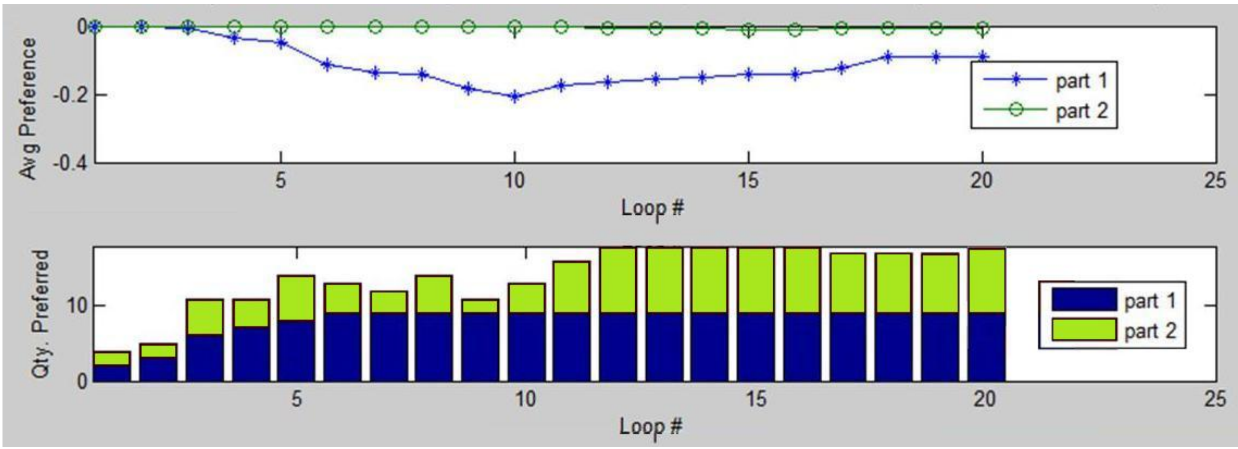


Figure 6.13: Test data showing the preference model improvements during the learning period, and a higher quantity of preferred designs being found.

through 10 iterations of the learning period. The bar chart shows that during the learning period, there is progressively more preferred designs in a numerical search that incorporates a preference-based model (part 1), as compared to a parallel search that does not incorporate preference at all (part 2).

This example, and test results, successfully demonstrated the use of all the steps of the design methodology to search through the vast number of possible combinations of design features and parameters (10,800 per iteration), and converge on a set of preferred designs, as shown in Figure 6.14. The vehicle features present in these designs are recognizable from the initial concept sketches (See Figure 6.7), however some of the combinations of features are new and were not previously considered by designers.

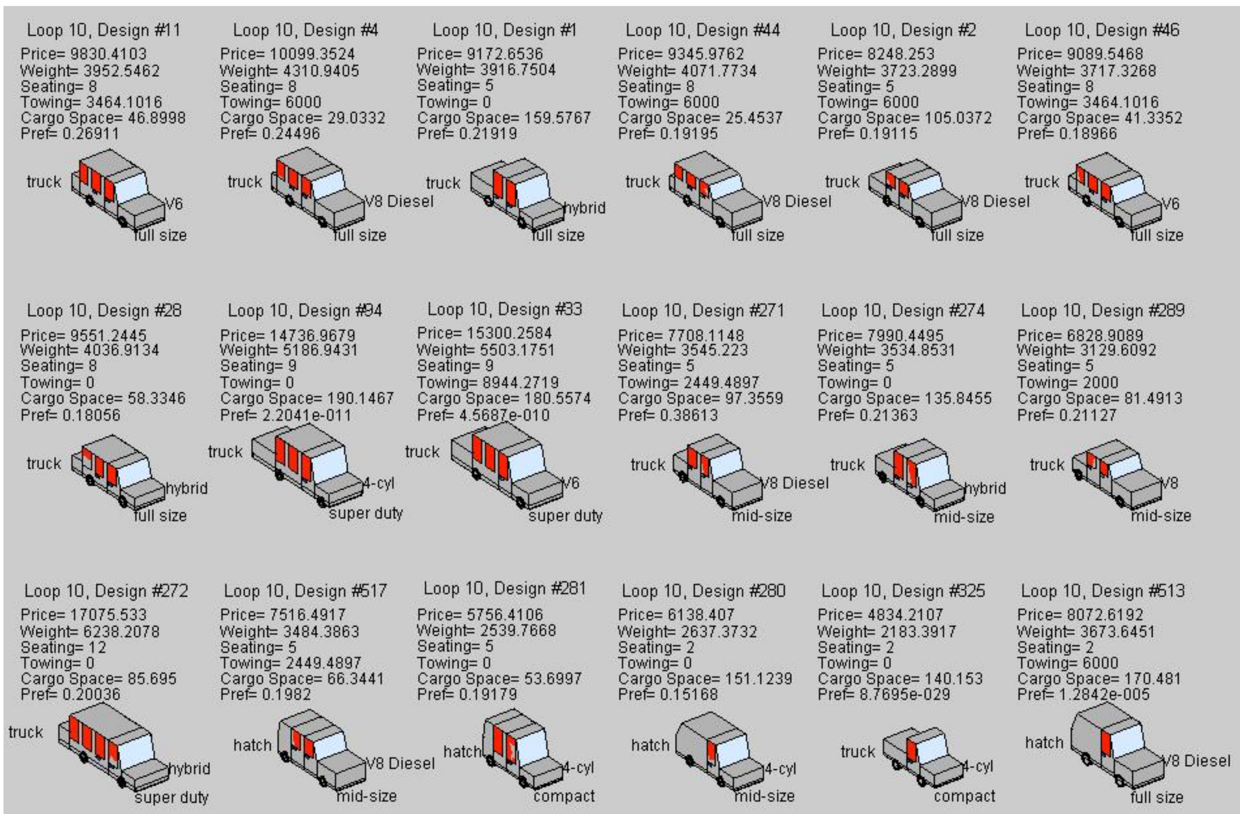


Figure 6.14: New, optimized vehicle designs automatically formed, consisting of new combinations of features and parameter values.

CHAPTER 7. CONCLUSIONS

This thesis has focused on improving the quality and quantity in the set of designs that is considered during conceptual design, by rapidly exploring for design possibilities and by incorporating human-based subjective evaluation into a computational search. To do this, a computationally-assisted design methodology was developed and formally organized into 11 steps, which was able to rapidly evaluate tens of thousands of designs per minute. Also, human-based subjective evaluation was captured to incorporate designer preference into the automated search.

The design examples began with initial sets of human-based manually-generated designs. The decomposed features of those designs were parameterized and used in a numerical search to find optimally performing designs according to physics-based models *and* preference-based models. It was shown how the new design methodology uses a statistics-based preference capture method to form a quantitative model of the subjective design decisions of a designer made during the manual evaluation of designs. This preference-based model, along with the physics-based models, was used in multiobjective optimization to guide the numerical search for designs that are *Pareto-optimal*, match the *preference* of the designer, and are *new combinations* of features and parameter values that may not have been found through manual methods.

The new computationally-assisted design methodology is able to harness computational power to evaluate *tens of thousands* of designs per minute, and still take advantage of the experience, intuition, and subjectivity of human designers. The relatively simple preference capture method used in this work parallels much of the learning methods being developed in the area of interactive evolutionary computing (IEC) research [31]. Future work related to this thesis could cross-functionally incorporate the newest IEC methods into the conceptual design process. Efforts to improve the effectiveness of the subjective evaluation done by designers could include improvements in the visualization of design composition and performance. The preference capture methods in this work also do not account for combinatorial effects or unequal weighting of preferred features and performance levels.

There exists the challenge of how to model performance of designs when unfamiliar combinations of features occur that are not covered by the existing models. These new combinations can either be treated as infeasible designs, or as an opportunity to develop new performance models potentially leading to the discovery of innovative products. It is likely that the application of this methodology will be most successful for design teams that repeatedly redesign the same type of products because they will have access to well developed physics-based models, or for the design of modular products. The methodology could also assist these same teams to more fully explore the design space and find designs that accomplish their existing design requirements, but with new combinations of features that had not previously been considered. Considering this, it seems that this methodology could be a useful conceptual design tool in industries such as consumer products, automotive, consumer electronics, recreational products, or any product that has the combination of qualitative design requirements and quantitative, engineered performance requirements. It would be more challenging to use this methodology when the design requirements are abstract and the goal is to redefine the overall functions of a product, rather than finding better performance through new features and subfunctions.

It was demonstrated that the design methodology can be used to more thoroughly search through an initial set of concepts for the best combination of features and parameters by using optimization methods. However, the contradiction of *convergence* upon a single design during a time in the design process that is intended for discovery and innovation, is one that must seriously be considered. If great products come from truly innovative ideas, then future work related to this thesis could include the development of methods to automatically inject new ideas and the models to evaluate the new ideas.

References

[1] Pahl, G., and Beitz, W., 1996. *Engineering Design: A Systematic Approach*. Springer.

[2] Ulrich, K., and Eppinger, S., 1995. *Product design and development*. McGraw-Hill New York.

[3] Ullman, D., 2002. *The Mechanical Design Process*. McGraw-Hill Science/Engineering/Math.

[4] Pugh, S., 1990. *Total design*. Addison-Wesley.

[5] Otto, K., 1995. "Measurement methods for product evaluation." *Research in Engineering Design,* **7**(2), pp. 86–101.

[6] Huang, G., and Mak, K., 1999. "Web-based morphological charts for concept design in collaborative product development." *Journal of Intelligent Manufacturing,* **10**(3), pp. 267–278.

[7] Stone, R., Wood, K., and Crawford, R., 1999. "Product architecture development with quantitative functional models." *Proceedings of the Design Engineering Technical Conferences*.

[8] Bryant, C., Stone, R., McAdams, D., Kurtoglu, T., and Campbell, M., 2005. "A computational technique for concept generation." *Proceedings of DETC2005, no. DETC05/DTM-85323*.

[9] Kopena, J., and Regli, W., 2003. "Functional Modeling of Engineering Designs for the Semantic Web." *IEEE Data Engineering Bulletin,* **26**(4), pp. 55–62.

[10] Bryant, C., Stone, R., McAdams, D., Kurtoglu, T., and Campbell, M., 2006. "A validation study of an automated concept generator design tool." *Proceedings of DETC/DTM2006, no. DETC2006-99489*.

[11] Bryant, C., Bohm, M., Stone, R., and McAdams, D., 2007. "An interactive morphological matrix computational design tool: A hybrid of two methods." *Proceedings of DETC/DTM2007, no. DETC2007-35583*.

[12] Hutcheson, R., Jordan Jr, R., Stone, R., Terpenny, J., and Chang, X., 2006. "Application of a genetic algorithm to concept variant selection." *Proceedings of DETC'06 (Design Engineering Technical Conferences)*, pp. 2006–99390.

[13] Barnum, G., and Mattson, C., 2009. "A Numerical Optimization Search Strategy for Exploring Morphological Charts." In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics,.

[14] Barnum, G., and Mattson, C., 2010. "A Computationally-assisted Methodology for Preference-guided Conceptual Design." In *Abstract submitted to the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, September 13 - 15, 2010, Fort Worth, Texas*, American Institute of Aeronautics and Astronautics,.

[15] Pugh, S., 1981. "Concept Selection-A Method that Works." *International Conference on Engineering Design,. Rome, Italy*.

[16] Suh, N., 1990. *The principles of design*. Oxford University Press New York.

[17] Al'tshuller, G., 1984. *Creativity as an Exact Science: The Theory of the Solution of Inventive Problems*. CRC Press.

[18] Pimmler, T., and Eppinger, S., 1994. *Integration Analysis of Product Decompositions*. Alfred P. Sloan School of Management, Massachusetts Institute of Technology.

[19] Qiu, S., Fok, S., Chen, C., and Xu, S., 2002. "Conceptual Design Using Evolution Strategy." *The International Journal of Advanced Manufacturing Technology,* **20**(9), pp. 683–691.

[20] Huang, G., Lee, S., and Mak, K., 2001. "Synchronised Web Applications for Product Development in the 21st Century." *The International Journal of Advanced Manufacturing Technology,* **18**(8), pp. 605–613.

[21] Huang, G., Lee, S., and Mak, K., 2003. "Collaborative product definition on the Internet: a case study." *Journal of Materials Processing Tech.,* **139**(1-3), pp. 51–57.

[22] Allen, K., and Carlson-Skalak, S., 1998. "Defining product architecture during conceptual design." *1998 ASME Design Engineering Technical Conference, Atlanta, GA, paper No. DETC98/DTM-5650*.

[23] Landon, M., and Balling, R., 1994. "Optimal packaging of complex parametric solids according to mass property criteria." *Journal of Mechanical Design,* **116**, p. 375.

[24] Yin, S., and Cagan, J., 2000. "An extended pattern search algorithm for three-dimensional component layout." *Journal of Mechanical Design,* **122**, p. 102.

[25] Mitchell, T., 1997. "Machine learning. WCB." *Mac Graw Hill*, p. 368.

[26] Norton, D., 2009. "Learning about creativity from an artificial artist." In *C&C '09: Proceeding of the seventh ACM conference on Creativity and cognition*, ACM, pp. 339–340.

[27] Dickerson, K. B., and Ventura, D., 2009. "Music recommendation and query-by-content using self-organizing maps." *Neural Networks, IEEE - INNS - ENNS International Joint Conference on,* **0**, pp. 705–710.

[28] Michalek, J., and Papalambros, P., 2002. "Interactive design optimization of architectural layouts." *Engineering Optimization,* **34**(5), pp. 485–501.

[29] Michalek, J., Choudhary, R., and Papalambros, P., 2002. "Architectural layout design optimization." *Engineering optimization,* **34**(5), pp. 461–484.

[30] Parmee, I., 2001. *Evolutionary and adaptive computing in engineering design*. Springer Verlag.

[31] Takagi, H., et al., 2001. "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation." *Proceedings of the IEEE,* **89**(9), pp. 1275–1296.

[32] Silverman, B., 1998. *Density estimation for statistics and data analysis*. Chapman & Hall/CRC.

[33] Wand, M., and Jones, M., 1995. *Kernel smoothing*. Chapman & Hall/CRC.

[34] Bowman, A., and Azzalini, A., 1997. *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*. Oxford University Press, USA.

[35] Balling, R., 2000. "Pareto sets in decision-based design." *Journal of Engineering Valuation and Cost Analysis,* **3**(2), pp. 189–198.

[36] Das, I., 1999. "A preference ordering among various Pareto optimal alternatives." *Structural and Multidisciplinary Optimization,* **18**(1), pp. 30–35.

[37] Kasprzak, E., and Lewis, K., 2000. "An approach to facilitate decision tradeoffs in Pareto solution sets." *Journal of Engineering Valuation and Cost Analysis,* **3**(2), pp. 173–187.

[38] Li, Y., Fadel, G., and Wiecek, M., 1998. "Approximating Pareto curves using the hyper-ellipse." In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper No. AIAA-98-4961*.

[39] Mattson, C., and Messac, A., 2003. "Concept selection using s-Pareto frontiers." *AIAA journal,* **41**(6), pp. 1190–1198.

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

[41] Park, K., and Grierson, D., 2002. "Pareto-optimal conceptual design of the structural layout of buildings using a multicriteria genetic algorithm." *Computer-Aided Civil and Infrastructure Engineering,* **14**(3), pp. 163–170.

[42] Tappeta, R., Renaud, J., Messac, A., and Sundararaj, G., 2000. "Interactive physical programming: tradeoff analysis and decision making in multicriteria optimization." *AIAA journal,* **38**(5), pp. 917–926.

[43] Wu, J., and Azarm, S., 2001. "Metrics for quality assessment of a multiobjective design optimization solution set.." *Journal of Mechanical Design(Transactions of the ASME),* **123**(1), pp. 18–25.

[44] Miettinen, K., 1999. *Nonlinear multiobjective optimization*. Springer.

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

[46] Steuer, R., 1986. *Multiple criteria optimization: Theory, computation, and application*. John Wiley & Sons, Inc., New York.

[47] Belegundu, A., and Chandrupatla, T., 1999. *Optimization concepts and applications in engineering*. Prentice Hall, New Jersey.

[48] Balling, R., 2003. "The Maximin Fitness Function; Multiobjective City and Regional Planning." *Proceedings of EMO,* **3**, pp. 1–15.

[49] Deb, K., 2001. *Multi-objective optimization using evolutionary algorithms*. Wiley.

[50] Kurfman, M., Rajan, J., Stone, R., and Wood, K., 2001. "Functional Modeling Experimental Studies." *Proceedings of DETC2001, DETC2001/DTM-21709, Pittsburgh, PA*.

[51] Clerc, M., 2006. *Particle swarm optimization*. ISTE London, UK.

[52] Kennedy, J., Eberhart, R., et al., 1995. "Particle swarm optimization." In *Proceedings of IEEE international conference on neural networks*, Vol. 4, Piscataway, NJ: IEEE, pp. 1942–1948.

[53] Dumitrescu, D., Lazzerini, B., Jain, L. C., and Dumitrescu, A., 2000. *Evolutionary computation*. CRC Press, Inc., Boca Raton, FL, USA.

[54] Van Veldhuizen, D., and Lamont, G., 2000. "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art." *Evolutionary Computation,* **8**(2), pp. 125–147.

[55] Mattson, C., Mullur, A., and Messac, A., 2004. "Smart Pareto filter: Obtaining a minimal representation of multiobjective design space." *Engineering Optimization,* **36**(6), pp. 721–740.

[56] "Ford Motor Company." *http://www2.showroom.fordvehicles.com/Showroom.jsp*.