2008-11-25

Product Family Design Using Smart Pareto Filters

Jonathan D. Yearsley
Brigham Young University - Provo

PRODUCT FAMILY DESIGN USING SMART PARETO FILTERS

by

Jonathan D. Yearsley

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

Brigham Young University

April 2009

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Jonathan D. Yearsley

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Christopher A. Mattson, Chair
Date	Spencer P. Magleby
Date	Jordan J. Cox

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Jonathan D. Yearsley in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date	Christopher A. Mattson Chair, Graduate Committee

Accepted for the Department

Larry L. Howell
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of
Engineering and Technology

ABSTRACT

PRODUCT FAMILY DESIGN USING SMART PARETO FILTERS

Jonathan D. Yearsley

Department of Mechanical Engineering

Master of Science

Product families are frequently used to provide consumers with a variety of appealing products and to help maintain reasonably low production costs for manufacturers. Three common objectives in the design of product families are used to balance the interests of both consumers and manufacturers. These objectives are to maximize (i) product performance, (ii) product distinctiveness as perceived by the consumer, and (iii) product commonality as seen by the manufacturer.

In this thesis, three methods are introduced that use multiobjective optimization and Smart Pareto filtering to satisfy the three objectives of product family design. The methods are progressive in nature and begin with the selection of product family members using Smart filtering and develop through the establishment of scale-based product platforms to the design of combined scale-based and module-based product platforms.

Each of the methods is demonstrated using a well-know universal electric motor example problem. The results of each method are then compared to a benchmark electric motor product family that was previously defined in the literature. Additionally, a pressure vessel example problem is used to further demonstrate the first of the three methods.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Christopher Mattson. His help has be invaluable during the research process as well as during the writing of each of the papers that together constitute this thesis. I would also like to thank my committee members, Dr. Spencer Magleby and Dr. Jordan Cox. Further, I would like to thank the Brigham Young University Mechanical Engineering Department that provided the funding for this work. Above all, I express my appreciation to my wife Brittany. I'm grateful for her patience and support throughout my undergraduate and graduate studies.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

In today's world marketplace, consumers increasingly demand variety, customization, and personalization in the goods they purchase [1]. For manufacturers, economically satisfying these demands is a notable challenge. The field of mass customization [2] has emerged, in recent years, in direct answer to these challenges. Within the realm of mass customization, a number of complementary tools have been developed including: computer-aided-engineering, design automation, and flexible manufacturing. Each of these tools further enables manufacturers to produce a truly endless set of customized products that near perfectly satisfy consumers' wants and needs.

However, because of limits in manufacturing technologies and processes, many products cannot be economically produced in a full customizable fashion. Take, for example, a laptop computer. A variety of screen sizes and storage capacities are available, but these product attributes are not customizable over a continuous range. Due to the complex processes involved in their production, a discrete set of sufficiently different screens and hard drives are made available instead.

As is the case with a laptop's screen and hard drive, many other products cannot be produced in a fully customizable fashion using today's technology. As an alternative to full customization, product families can provide consumers with a sufficient, albeit limited, variety of appealing products while helping manufacturers maintain reasonably low production costs through increased economies of scale and scope [3]. The objective of this thesis is the development of an approach that aids designers in determining product family size, members, and platform, including the identification of modular components and scalable parameters to be used in the design of a combined module-based and scale-based product family.

1.2 Background and Literature Survey

When designing product families, three objectives are considered to balance the interests of both consumers and manufacturers. These goals are to maximize (i) product performance, (ii) product distinctiveness as perceived by the consumer, and (iii) product commonality as seen by the manufacturer [4, 5].

Product families are built on two primary types of platforms: module-based platforms and scale-based platforms [6]. A module-based platform is the foundation for a collection of related products that have differing functions through the addition or subtraction of modules. A scale-based platform is the foundation of related products that have differing function through the scaling of non-platform design features.

In the literature a variety of approaches exist for the optimization of both scale-based and module-based product families. Many of these approaches use sequential quadratic programing [7, 8], genetic algorithms [9–11], generalized reduced gradients [12, 13], or orthogonal arrays [14, 15] in determining a solution. In most cases, scale-based and module-based design are treated independently, although the two types of platform design are not mutually exclusive. Only two cases were identified where scale-based and module-based design were addressed in conjunction with each other [16, 17].

Many scale-based platform design methods focus on identifying the optimal set of and optimal values for the design variables that define the product platform. Traditionally, to optimize a scale-based product family the following primary tasks are completed.

(1) Determine the number of and identify product family members through *market segmentation*.

(2) Identify the variables that are held constant for all products in the product family; the platform variables.

(3) Determine the most suitable value of each platform variable.

(4) Determine the value of each non-platform or scalable variable.

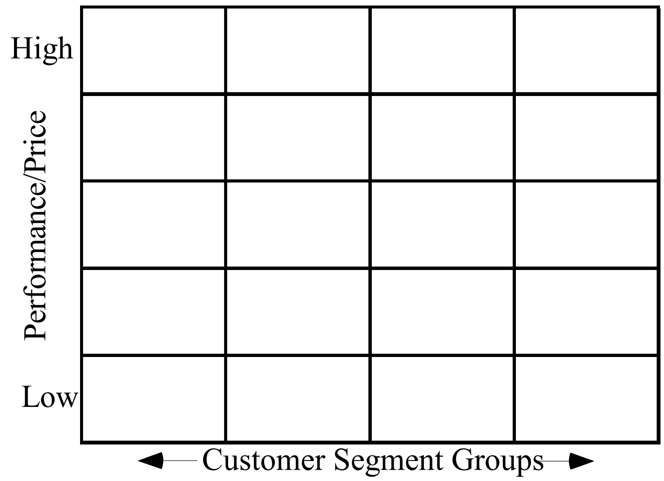


Figure 1.1: Traditional market segmentation grid

Module-based platform design methods have traditionally focused on the combinatorial nature of modular design. The following three primary tasks are often completed in the design of a module-based product platform.

(1) Determine the number of and identify product family members through *market segmentation*.

(2) If not predefined, define module attributes using scale-based optimization.

(3) Determine the optimal combination of modules in the design of each product family member.

As is noted in the process descriptions above, the first step for either scale-based or module-based design is, traditionally, the use of market segmentation to identify product family members. When using market segmentation, a grid is constructed by first determining principal customer groups, then by subdividing each group using product performance/price gradations [18]. Thus used, market segmentation discretizes the market to determine the number of product family members and identify the members themselves. A typical segmentation grid is shown in 1.1.

To ensure that each member of the optimized product family lies within its respective product performance envelope, as defined through market segmentation, equality constraints are often defined so that each product will possess certain performance characteristics [19, 20]. Performance characteristics that are not captured in the constraints are made design objectives and each product family member is individually optimized. The use of market segmentation for defining product family members produces encouraging results. However, this approach depends on a company's ability to identify and segment the market, a task to which complete disciplines and full departments within a company are often dedicated.

However, when sufficient market data is unavailable or is not easily obtained or distilled by the company developing the product family, traditional segmentation methods become difficult to implement. For example, entrepreneurs and small to mid-sized companies may not have a dedicated marketing department or the financial resources needed to purchase market research from an outside firm. When planning to launch a series of products built on a common platform, these small companies may struggle to gather and process the market data needed to construct the segmentation grid traditionally used to identify product family members. As an alternative to traditional segmentation, the use of Smart Pareto filters as a method for selecting product family members is introduced in this thesis. Thus used, Smart Pareto filtering leverages the designer's knowledge of the product area and/or customer interests when selecting members of a product family. Importantly, Smart Pareto filtering is not intended to replace market segmentation in every application. It is intended, however, to provide a systematic method for family member selection when traditional segmentation cannot be easily used.

Additional methods that address the intricate and important tradeoff that often exists between product commonality, functional performance, market performance, manufacturing costs, and revenue have been developed to aid designers in defining a product family. Nested Logit Demand Modeling [21] and leveraging existing market models [22] are two approaches for considering the effect that increased commonality may have on the market performance of product family members. The added power of these methods to consider the effect of commonality decisions on market performance is valuable. These methods, however, also require extensive knowledge of the product family's potential market niche as well as competing products.

Each of the individual papers that constitute this thesis contain their own literature survey that provides additional background information on the design of product families.

1.3 Research Approach

The research included in this thesis was developed through the preparation of three recent publications [23–25]. These publications constitute the following three chapters. In each of the following chapters, variations on a common design approach, including the use of Smart Pareto Filters in the selection of product family members, are presented for the design of product families. The common approach is summarized in the following three step process:

(1) Objective performance of each product family member is maximized through the use of multiobjective optimization.
(2) Product distinctiveness is ensured by using Smart Pareto filters for the selection of product family members.
(3) One of three distinct methods of establishing product platforms is used to maximize product commonality.

Chapter 2 introduces the use of Smart Pareto filtering for the selection of product family members [23]. The method begins by surveying the family's multiobjective design space and identifying a discrete representation of the Pareto frontier. Importantly, the Pareto frontier captures the tradeoff among all optimal product configurations. A Smart Pareto filter is then used to reduce the Pareto frontier to the spanning set of product configurations to include in the product family. Next, the design variables that are best suited as platform variables and those that are best suited as scalable variables are identified using a previously published method. Finally, the product family, sharing a common platform, is compared to the set of individually optimized products to show the change in performance resulting from the implementation of the product family.

In Chapter 3, a Pareto filter that concurrently considers both objective and variable spaces is used to identify a product family where each product in the family possesses optimal and distinct performance characteristics [24]. Simultaneously, the filter also identifies features that can be shared by the products in the family. The filter functions as follows: (1) Given a set of Pareto solutions, a starting design is identified and classified as the first member of the product family. (2) A region of insignificant tradeoff is constructed about the starting design in objective space. Any design located within the region of insignificant tradeoff is considered insignificantly different from the starting design and is eliminated from the set of candidate product family members. (3) Each remaining candidate product family member is ranked according to the effect that its addition to the product family would have on the variation among design variables. Specifically, the design resulting in the lowest calculated standard deviation to mean ratio, summed across all design variables, is selected as the next member of the product family. (4) A region of insignificant tradeoff is constructed about this design and any designs located within this space are removed from the set of candidate product family members. Steps 3 and 4 are then repeated until all points have either been included as a member of the product family or identified as being insignificantly different from at least one product family member. Considering all products in the family, if for any design variable the variation from product to product in the family is sufficiently small, then that variable is identified as well suited to become common to the entire family.

Chapter 4 introduces an interactive framework that is used in conjunction with concurrent Smart Pareto filtering in variable and objective spaces in the design of a combined scale-based *and* module-based product platform [25]. In the first step of this interactive process, the designer uses physical decomposition techniques to identify the major components that comprise the finished products. Second, the designer identifies and summarizes, in matrix form, the relationships that exist between each design variable and each product component. Third, multiobjective optimization is used to identify a set of many designs that are considered candidate product family members. Fourth, Smart Pareto filtering is used to select product family members from among the candidate set previously identified. A scale-based product platform is simultaneously established using concurrent Smart Pareto filtering and the component/variable relationships identified in the second step of this process. Finally, the component/variable relationships are further used to determine which product components are best suited to become modules and to determine the subset of product family members that will use each module. In conjunction with the establishment of each module, a designer specified sequence of optimization routines is used to ensure that the three objectives of product family design, as noted above, are satisfied. As a result, a combined scale-based and module-based product platform is established.

The three publications that constitute the following chapters collectively satisfy the objectives of this thesis. Chapter 2 presents Pareto frontiers as candidate sets of product family members and introduces the use of Smart Pareto filters for the selection of family members. Chapter 3 builds upon the use of Smart Pareto filters by concurrently searching design variable space, allowing for the simultaneous selection of product family members and establishment of the product platform. Chapter 4 then applies concurrent Smart Pareto filtering in the design of product families built upon a combined scale-based and module-based platform.

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. Within this final chapter, the applications and limitations of each of the methods presented throughout the body of the thesis are discussed.

Chapter 2

Product Family Design using a Smart Pareto Filter

2.1 Abstract

The design of product families requires that the family members be identified and the optimal product platform be defined. In this paper, we present a Pareto filtering method that can be used to determine the number of members to include in a product family and identify the members themselves. The proposed method, coupled with common approaches for defining the product platform, provides the designer with a new and alternative framework for designing a product family. This method, which is not based on traditional market segmentation data, is of particular use when such data is unavailable or is not easily gathered or distilled by the company designing the product family. The method begins by surveying the family's multiobjective design space and identifying a discrete representation of the Pareto frontier. Importantly, the Pareto frontier captures the tradeoff among all optimal product configurations. A designer-controlled filter is then used to reduce the Pareto frontier to the spanning set of product configurations to include in the product family. This spanning set, known in the literature as a Smart Pareto set, includes the members needed to make a product family sufficiently diverse and provides adequate representation of performance tradeoffs. As a near-final step in establishing the product family, the design variables that are best suited as platform variables and those that are best suited as scalable variables are determined using previously published methods. Finally, the product family, sharing a common platform, is compared to the set of individually optimized products to show the change in performance resulting from the implementation of the product family. The result is a spanning set of products, designed on a common platform, that together

comprise a product family. Well-known case studies are used to demonstrate the method: universal electric motor design and pressure vessel design.

2.2 Nomenclature

n_p	Number of products in product family
n_x	Number of design variables
n_μ	Number of design metrics (objectives)
x	Vector of design variables
μ	Vector of design metrics (objectives)
g	Vector of inequality constraints
h	Vector of equality constraints
δ_j	Maximum allowable variation in design variable x_j, across the product family
x_j^*	Vector of optimal values for design variable x_j
Δx_j^*	Maximum difference among all values in x_j^*, across the product family
J_i	Aggregate objective function for product i
F	Average fitness of product family
R	Pressure vessel radius
L	Pressure vessel length
T_s	Thickness of pressure vessel shell
T_h	Thickness of pressure vessel head
V	Volume of pressure vessel
P	Maximum safe pressure of vessel
Γ	Selling price of pressure vessel
N_a	Number of wire turns on armature
N_f	Number of wire turns on field, per pole
t	Thickness of stator
I	Electric current
L_s	Stack length
A_a	Armature wire cross-sectional area

A_f	Field wire cross-sectional area
r	Outer radius of stator
η	Efficiency
M	Motor mass
T	Motor torque
H	Magnetizing intensity
P_o	Gross mechanical power output
P_i	Electrical power input
M_s	Stator mass
M_a	Armature mass
M_w	Windings mass
K	Motor constant
ϕ	Magnetic flux

2.3 Introduction and Literature Survey

The current world marketplace demands variety in consumer and commercial goods. In order to be profitable and competitive in a market that emphasizes variety, manufacturers are striving to produce, at mass production efficiencies, products that appeal to various consumers. *Product families* can be leveraged to maintain or improve development and production efficiencies while increasing product variety and as a result attract a wide range of customers. A product family can be defined as a group of related products that are built upon a common product platform and that balance product commonality and performance diversity [4]. A product platform is defined in general terms as the basis of a product family; the platform consists of features that are common to products within the family. The literature specifically includes shared components, modules [26] and/or production processes [27] in a product platform definition. By sharing components, processes, and other features among products that are otherwise disparate, product families can decrease design and manufacturing costs by using previously designed components and by economies of scale [5].

Product families are built on two primary types of platforms: module-based platforms and scale-based platforms [6]. A module-based platform is the foundation for a collection of related products that have differing functions through the addition or subtraction of modules. A scale-based platform is the foundation of related products that have differing function through the scaling of non-platform design features.

To aid the design of scale-based product families, a number of optimization methods have been developed [6]. Many of these methods have focused on identifying the optimal set of and optimal values for the product platform variables that define the platform. The objective of these methods is to create a family of sufficiently diverse products that possess a high level of commonality and only minimal performance compromises, when compared to individually optimized products. Traditionally, to optimize a scale-based product family the following primary tasks are completed.

(1) Determine the number of and identify product family members through *market segmentation*.

(2) Identify the variables that are held constant for all products in the product family; the platform variables.

(3) Determine the most suitable value of each platform variable.

(4) Determine the value of each non-platform or scalable variable.

Market segmentation, an important part of the first step in the traditional approach, constructs a market grid that can be used to identify possible product family members. Market segmentation first determines principal customer groups, then subdivides each group using product performance/price gradations [18]. Thus used, market segmentation discretizes the market to determine the number of product family members and identify the members themselves. The set of products identified though segmentation make up the product family that is subsequently optimized in steps two through four of the traditional approach.

To ensure that each member of the optimized product family lies within its respective product performance envelope, as defined through market segmentation, equality constraints are often defined so that each product will possess certain performance characteristics [19, 20]. Performance characteristics that are not captured in the constraints are made design objectives and each product family member is individually optimized. The use of market segmentation for defining product family members produces encouraging results. However, this traditional approach depends on a company's ability to identify and segment the market. A task to which complete disciplines and full departments within a company are often dedicated.

We pause now to make an important comment regarding the identification of product family members. Both the market segmentation approach and the method presented herein identify the product family members only in an approximate way. Each member is only approximately identified because its performance characteristics will change, often in a small way, with the establishment of the product platform. For this reason the product family members are not completely identified until the platform has been established.

Additional methods that address the intricate and important tradeoff that often exists between product commonality, functional performance, market performance, manufacturing costs, and revenue have been developed to aid designers in defining a product family. Nested Logit Demand Modeling [21] and leveraging existing market models [22] are two approaches for considering the effect that increased commonality may have on the market performance of product family members. The added power of these methods to consider the effect of commonality decisions on market performance is valuable. These methods, however, also require extensive knowledge of the product family's potential market niche as well as competing products.

In this paper, we propose an alternative method for designing a product family – one that can be used when traditional market data is unavailable or is not easily obtained or distilled by the company developing the product family. Such an alternative method may be of particular interest to entrepreneurs or small to mid-sized companies planning to launch a series of products built on a common platform but that do not have adequate resources to gather/process the market data. Rather than identifying product family members by constructing a market segmentation grid, the proposed method leverages the designer's knowledge of the product area and/or customer interests to select family members.

The method described herein begins by surveying the family's multiobjective design space and identifying a discrete representation of the Pareto frontier, which is defined as the set of non-dominated design solutions. Depending on the design at hand, these solutions may represent the optimal tradeoffs among size, weight, cost, number of passengers, towing capacity, fuel type, or any number of additional performance considerations and importantly, can be considered a set of candidate family members. Further, the set of non-dominated design solutions is of particular interest for product family design because it can be used to identify disparate designs that cover a wide range of performance envelopes thereby aiding the designer in finding a set of products that appeal to a wide range of customers.

In the next step of the proposed method, a filter based on designer knowledge of the product area and/or customer interests is used to reduce the discrete Pareto frontier to a spanning set, known in the literature as a Smart Pareto set [28], where each point in the set is a member of the product family. Because the filter only selects points that are sufficiently diverse, the products selected to be members of the family are distinct, thereby minimizing performance overlap within the product family. Next, the product platform is defined by identifying the design variables that are best suited as platform variables as well as those that are best suited as scalable variables. This is done using existing methods, one of which is briefly described in the next section. The result is a product family that shares a common platform and that is comprised of a spanning set of family members that provide adequate variation in performance characteristics. The method concludes by comparing the product family to the Smart Pareto set of individually optimized products to show the performance compromises that result from the implementation of the product family.

The remainder of this paper is presented as follows. A preliminary review of the simple platform method used in the example problems is included in Section 2.4. In Section 2.5, the proposed Pareto filtering method is presented in its theoretical form. Examples illustrating the application of the method are presented in Section 2.6, with our concluding remarks given in Section 2.7.

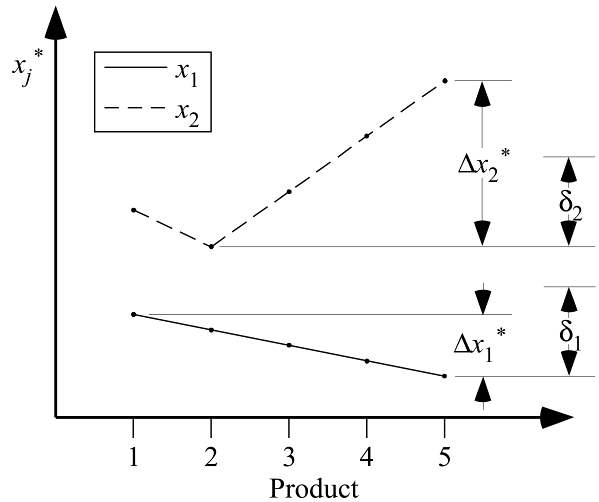


Figure 2.1: Identifying platform variables using each variable's threshold of variation

2.4 Technical Preliminary: A Product Platform Definition Approach

This section briefly presents a typical method for defining the product platform [29]. It is presented here for completeness, as the method is used in both of the example problems included in Section 2.6. The purpose of the method described in this section is to identify which variables should be fixed across the complete product family, and what their values should be.

Figure 2.1 depicts the framework for classifying the variables. Along the horizontal axis of the plot, a discrete space represents the products in the product family; product 1, product 2, and so on. Along the vertical axis of the plot are the variable values. The solid line represents the variable x_1, while the dashed line represents x_2. The purpose of this plot is to illustrate how the variables differ from product to product within the family.

A maximum allowable difference (δ_j) in the values of a single design variable (j), across all products in the product family is provided by the designer for the identification of platform variables. This maximum allowable difference is called the *threshold of variation*. In vector form, the threshold of variation set is summarized by Equation 2.1.

$$\delta = [\delta_1, \delta_2, ..., \delta_{n_x}] \tag{2.1}$$

Each design variable's threshold of variation is used to determine if that variable is well suited as a platform variable or if it is better suited as a scalable variable. For each design variable x_j where $j = 1,...,n_x$, the largest possible difference is taken between the elements of the vector x_j^* and the difference is then compared to the corresponding threshold of variation, δ_j. Mathematically, the process of comparison is defined as follows:

If:

$$\underset{j}{\delta} > \Delta x_j^* \tag{2.2}$$

Where:

$$\Delta x_j^* = \text{maximum}(x_j^*) - \text{minimum}(x_j^*) \tag{2.3}$$

Then: x_j is best suited as a platform variable.

From the comparison shown in Figure 2.1, it is seen that design variable x_2 is best suited as a scalable variable while x_1 is well suited as a platform variable, because Δx_1^* lies within the limits of the envelope of allowable variation defined by δ_1.

Once the platform variables are identified, the value of each one must be determined in order to fully define the product platform and finalize family member identification. In defining the product platform, the method explores platform variable values at the maximum, minimum, and mean values of each platform variable x_j^*. If more than one of these values is found to satisfy all design constraints, the value that introduces the least amount of change, as compared to the individually optimized products, is selected. It is important to note that the establishment of the platform will likely move one or more members off the Pareto frontier. This effect will generally become more extensive as the product family becomes more common (i.e., more variables become platform variables). This is a primary conflict to be mitigated during product family design; thereby achieving maximum commonality with minimal deviation from performance optimality.

2.5 A Framework for Determining the Number of Family Members and Identifying the Members Themselves

In this section, we introduce a design framework that can be used to determine the number of members to include in a product family, and identify the members themselves. The method requires that: (i) a multiobjective optimization problem can be formulated that captures the objectives of the product family, (ii) the problem can be solved for a set of solutions comprising a Pareto frontier, and (iii) the designer has sufficient experience in the product area to be able to specify *intuitive parameters* that define design distinctiveness (these parameters are defined shortly). The proposed framework is presented according to the flow diagram of Figure 2.2 and is comprised of the following steps:

Step 1 Develop a multiobjective optimization statement by identifying the key performance objectives and constraints for the product family. From the optimization statement, generate a discrete Pareto frontier representation of the objective tradeoffs.

Step 2 Identify a spanning set of products that represent the multiobjective design space by using a Smart Pareto filter. The spanning set of Pareto designs consists of product family members and each is notably distinct from any other in the Smart set.

Step 3 Define the product platform by first identifying the design variables that are best suited as platform variables and then by determining the best-fit value for each platform variable.

Step 4 Identify the performance compromise (deviation from Pareto frontier) that results from the implementation of the product family by comparing the Smart Pareto set of individually optimized products to the products in the product family.

The subsections that follow present steps 1, 2, and 4 in detail. Recall that step 3 was presented in Section 2.4.

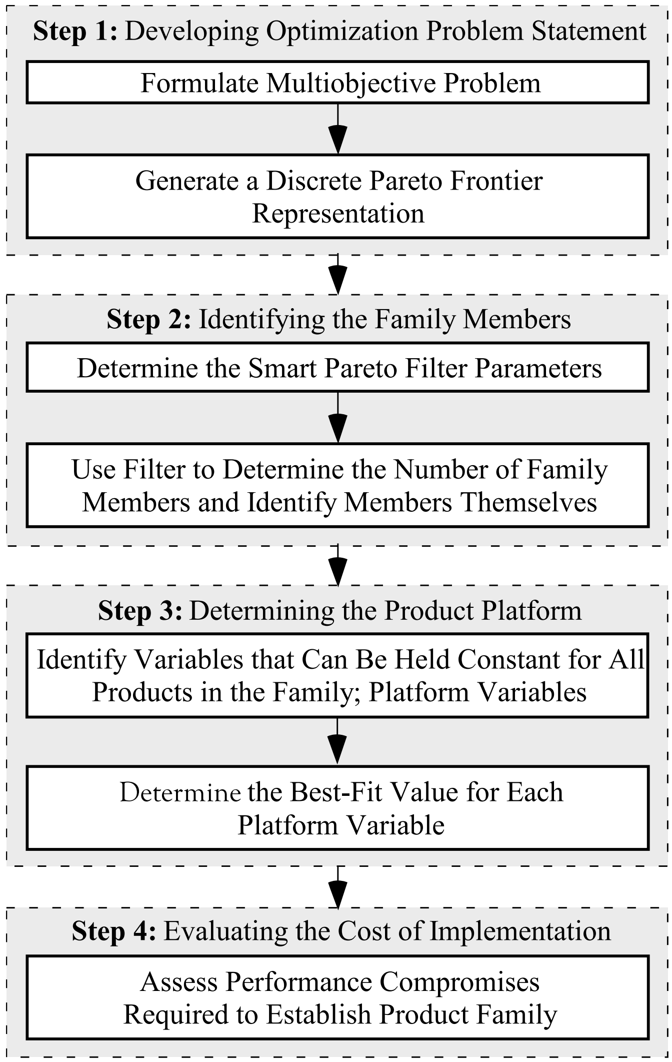


Figure 2.2: Pareto filtering approach to product family design

2.5.1 Multiobjective Optimization Problem

In product family design, multiobjective optimization is often used to evaluate the tradeoff that exists between the conflicting objectives that define product performance [30, 31]. We note that although multiobjective problems are not used exclusively, most product family design problems can be formulated as a multiobjective problem. Mathematically, a multiobjective optimization problem can be defined as shown in problem 1.

Problem 1 (P1): Generic Multiobjective Optimization

$$\min_{x} \mu(x) \tag{2.4}$$

subject to:

$$g(x) \leq 0 \tag{2.5}$$

$$h(x) = 0 \tag{2.6}$$

$$x_l \leq x \leq x_u \tag{2.7}$$

Multiobjective optimization problems, such as *P1*, are of interest in the selection of product family members because they result in many *Pareto solutions*. A Pareto solution is one where any improvement in one design objective can only occur at the expense of at least one other objective [32] and the complete set thereof comprise the Pareto frontier. Importantly, the Pareto frontier is a representation of the tradeoffs between conflicting design objectives. Figure 2.3(a) shows a Pareto frontier for a bi-objective problem. The Pareto frontier is of interest in the identification of product family members because it consists of every optimal combination of product performance objectives that could be included in the product family. Together, these designs comprise a set of candidate product family members. In general we don't identify an infinitely large set of designs that complete the frontier but rather a discrete set that adequately represents the frontier.

2.5.2 Using the Smart Filter to Determine the Number of Product Family Members

Through the generation of the Pareto frontier, candidate product family members are identified. Mattson et al. [28] observe, however, that many designs, corresponding to points along the Pareto frontier, are practically indistinguishable from one another. Nelson et al. [31] note that as products in a family become excessively common, the effectiveness of the product family decreases as the performance characteristics of the products within the family become less distinct. The method purposed in this paper identifies a set of family members that balances product commonality, performance, and distinctiveness by

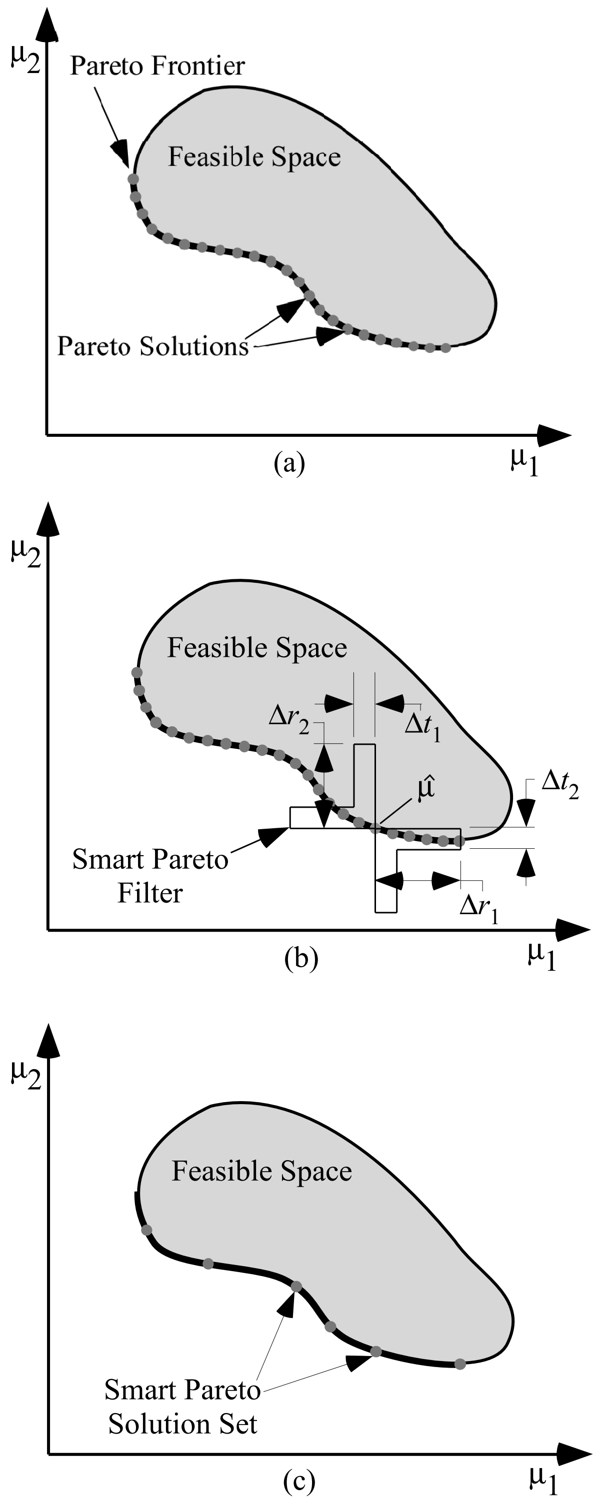


Figure 2.3: (a) Discrete representation of Pareto frontier (b) 2-D Smart Pareto filter (c) Smart Pareto set of solutions corresponding to family members

identifying a spanning set of Pareto solutions, including only those points that correspond to designs of sufficiently different, yet optimal, product performance.

A Smart Pareto filter is used to reduce the Pareto frontier to a minimal set of distinct designs, called the Smart Pareto set [28]. Smart Pareto filters perform pairwise comparisons of designs in a Pareto set. When the comparison concludes that the points are overly similar, one is removed from the set. The pairwise comparison, and consequently the Smart Pareto filter, is based on two designer defined parameters, Δt and Δr – as shown in Figure 2.3(b), for the bi-objective case. Specifically, Δt and Δr define a region called the *Region of Insignificant Tradeoff*. In Figure 2.3(b), the region of insignificant tradeoff is the space inside the geometric shape centered on the point labeled $\hat{\mu}$. Any point inside the region of insignificant tradeoff is deemed to be not significantly different than $\hat{\mu}$. Thus, the filter removes any point within the region of insignificant tradeoff, and defines the next of the remaining points in the set as $\hat{\mu}$ and repeats.

To execute the filter, the designer must be able to specify Δt and Δr. The two parameters can be understood by the following; when comparing two Pareto points (or designs), any difference in μ_i that is less than Δt_i is considered to be insignificant. As such, one of the two designs being compared should be removed since it is insignificantly different – at least in one objective. When, however, the difference in μ_i, between two points, is within Δt_i, but the change in another objective (μ_j) is significantly large (greater than Δr_j), then the design should not be removed. In other words, any change in μ_j that is larger than Δr_j is deemed significant – regardless of how small the change in another objective is.

The originally published Smart Pareto filter configuration [28], described above, performs well for problems where a notable *increase* or *decrease* in an objective value are of equal interest. However, when cost, for example, is included as an objective and all other objectives show insignificant tradeoffs, a notable decrease in cost is of interest but a notable increase in cost is of no interest, regardless of magnitude. Improving the general usefulness of the Smart Pareto filter, we have modified the filter to allow the designer to specify different levels of notable change in an objective, for increasing *and* decreasing objective values respectively. In other words, the Smart Pareto filter thus modified preferentially accepts an

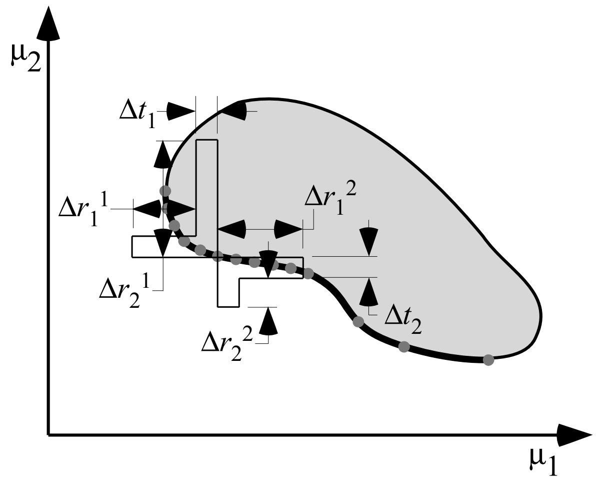


Figure 2.4: Smart Pareto filter with unequal levels of objective notability

objective change in one direction over the other. This modification allows for the correct treatment of objectives such as noise level, efficiency, and weight. The difference in levels of notable change for increasing and decreasing values of μ_2 can be seen by comparing Δr_2^1 and Δr_2^2 in Figure 2.4.

To provide the designer with the increased flexibility in characterizing the needed product family, we use separate sets of Δt_i and $\Delta r_i = \Delta r_i^1, \Delta r_i^2, ..., \Delta r_i^{n_\mu}$ for each objective and perform the filtering. We execute the filter on a non-normalized set of data so as to make the declaration of Δt_i and Δr_i physically meaningful. When determining the magnitude of the filter parameter values, it is helpful to consider each parameter with respect to market preferences. Δt_i represents a small change in objective performance that would be considered insignificant by consumers. Δr_i represents a larger change in objective performance that consumers would consider noteworthy in defining product variants regardless of insignificant changes in other objectives. The magnitude of each parameter should be chosen to reflect the knowledge available about these market preferences as closely as possible. The filter algorithm is detailed in Mattson et al. [28]

The designs in the Smart Pareto set provide a minimal representation of candidate members of the product family as is shown in Figure 2.3(c). From this we conclude that the Smart Pareto set of designs adequately covers the required product performance range, with minimal intra-family product competition. The Smart Pareto set, therefore, consists of the spanning set of needed product family members. Each design in the Smart Pareto set becomes a member of the product family, thereby determining the number of members to include in the product family as well as identifying the members themselves.

In using a Smart Pareto filter for the identification of product family members, it is important to note the following. We only consider designs lying on the Pareto frontier because any other design can be improved in every objective. The selection of product designs that will become members of the product family, from among the Pareto optimal solution set, is a somewhat subjective process that depends the filter starting point, as described in Mattson et al. [28]. In other words, the number of members determined to be included in the product family and the members themselves that are identified may vary in a minor way as the filter starting point is changed.

By nature of the Smart Pareto filter and the multiobjective optimization, the products that make up the Smart Pareto set identified in this section have mathematically optimal performance and an adequate level of distinctiveness. This set, consisting of product family members, must now be evaluated to determine the common platform upon which the product family is to be built. Many methods for establishing scale-based product platforms have been developed and could be implemented at this point to further define the product family [6]. For simplicity, the method presented in the Section 2.4 is used in the example problems that follow.

2.5.3 Characterizing the Performance Cost of Product Family Implementation

After defining a product platform using one of many available platform approaches, we seek to classify the performance change resulting from the establishment of the product family. To do this, the Smart Pareto set of individually optimized products and the platform-based product family are compared based on objective performance. In the establishment of a common platform, a number of performance changes can occur. A single design may be unaffected by the platform, it may decrease in performance, or it may shift to another Pareto optimal point. A set of designs may also become excessively similar through the implementation of the product platform. To eliminate products that may have become undesirable due to the establishment of the product platform, the designer may check the

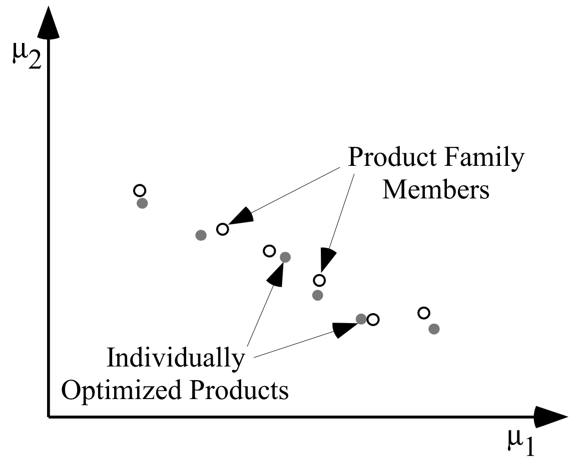


Figure 2.5: Performance change of product family due to platform implementation

Pareto optimality of the product family members and/or use a second iteration of the Smart Pareto filter to remove excessively similar products.

The change in family performance can be evaluated numerically by calculating a *family fitness* value. The family fitness value is an average measure of objective function value across the entire product family and is calculated using an aggregate objective function as shown in Equation 2.8. Note that this measure of family fitness exclusively considers the effect that establishing a platform has on the average objective performance of the family as a whole and does not consider commonality or diversity in its calculation.

$$F = \sum_{i=1}^{n_p} (J_i)/n_p \tag{2.8}$$

In the case of a minimization problem, a smaller family fitness value corresponds to a higher average performance for the product family. Figure 2.5 shows the comparison of a bi-objective product family and the Smart Pareto set of products from which the product family was derived. In the section that follows, two well-known case studies are examined using the proposed framework.

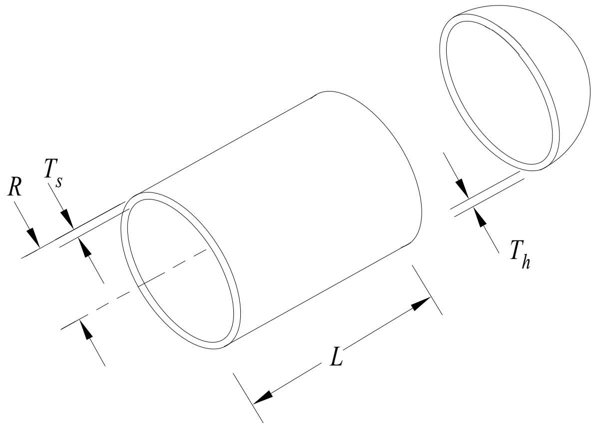


Figure 2.6: Pressure vessel schematic

2.6 Example Problems

To illustrate the application of the proposed method, two example problems are included in this section. The first problem is the design of a product family of pressure vessels. The second problem is the design of a family of universal electric motors. Both example problems are taken from a testbed of product family optimization problems [33] and have been reformulated as multiobjective problems.

2.6.1 Pressure Vessel Example Problem

The example that follows shows the application of the proposed method in the design of a family of pressure vessels. Recall that the method requires that a meaningful optimization problem be formulated, that the problem be solved for set of Pareto solutions, and that the designer specify parameters that define design distinctiveness (Δt and Δr). Specifically, in this example, we show that the number of product family members can be determined and that the members themselves can be identified using the proposed approach. Figure 2.6 shows a schematic of the generic pressure vessel. Problem 2 details the optimization problem statement used.

Problem 2 (P2): Multiobjective Optimization of a Family of Pressure Vessels

$$\min_{x}[-V(x) \quad -P(x) \quad \Gamma(x)]^T \tag{2.9}$$

subject to:

$$V - 30\ \text{m}^3 \leq 0 \tag{2.10}$$

$$10\ \text{m}^3 - V \leq 0 \tag{2.11}$$

$$P - 30\ \text{MPa} \leq 0 \tag{2.12}$$

$$10\ \text{MPa} - P \leq 0 \tag{2.13}$$

$$0\ \text{m} \leq R \leq 1.5\ \text{m} \tag{2.14}$$

$$0\ \text{m} \leq L \leq 7\ \text{m} \tag{2.15}$$

$$0.0063\ \text{m} \leq T_s \leq 0.0762\ \text{m} \tag{2.16}$$

$$0.0063\ \text{m} \leq T_h \leq 0.0762\ \text{m} \tag{2.17}$$

where:

$$x = [R, L, T_s, T_h] \tag{2.18}$$

$$V = \pi R^2 L + 4/3\pi R^3 \tag{2.19}$$

$$P = \text{minimum}\left[\frac{S_y T_s}{R + 0.06T_s}, \frac{10 S_y T_h}{5R + T_h}\right] \tag{2.20}$$

$$C_m = 2\pi\rho\left(C_s R T_s L + C_h R^2 T_h + C_{s_r} T_s R (L_{s_r} - L)\right) \tag{2.21}$$

$$C_w = 2\pi\rho(2/9 C_{w_m} T_s^2 L + 4/9 C_{w_m} \pi T_s^2 R) \tag{2.22}$$

$$\Gamma = 0.673(C_m + C_w) + 2700 \tag{2.23}$$

The fixed parameters for this problem are material density, $\rho = 7800\ \text{kg/m}^3$; cost of shell material, $C_s = \$0.80/\text{kg}$; cost of head material, $C_h = \$2.00/\text{kg}$, cost of raw plate material, $C_{s_r} = \$0.30/\text{kg}$; length of raw material, $L_{s_r} = 7$ m; and cost of welding material, $C_{m_w} = \$15.00/\text{kg}$.

Solving *P2* results in a discrete representation of the Pareto frontier, with each Pareto solution representing a candidate family member. Because each Pareto solution represents a potential product family member, it is important that the discrete Pareto frontier provide adequate and even representation of the entire design space so as to not ignore large groups of possible product family members. An ε-constraint method results in an

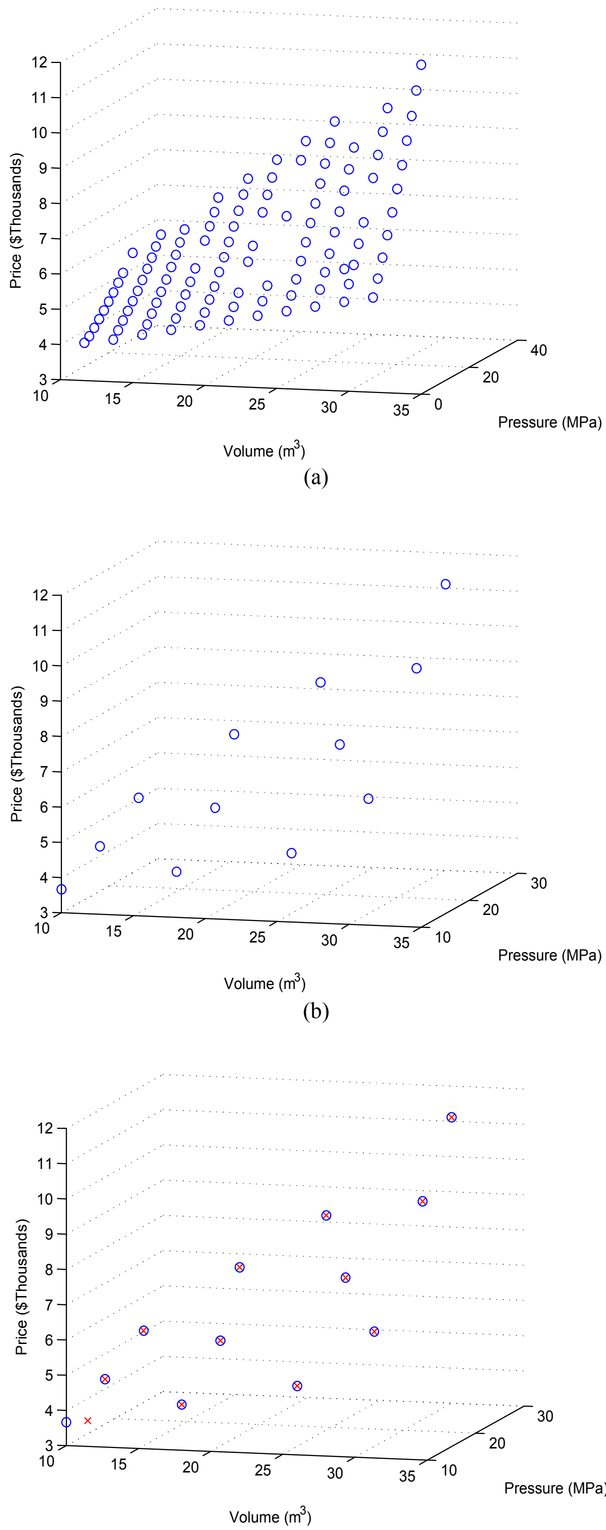
Table 2.1: Pressure vessel product family members before platform is selected

R	L	T_s	T_h	V	P	Γ
0.68	5.98	0.0063	0.0063	10	10.00	3.66
0.64	7.00	0.0108	0.0063	10	18.00	4.27
0.64	7.00	0.0156	0.0077	10	26.00	5.04
0.80	7.00	0.0210	0.0103	16	27.95	6.79
0.84	7.00	0.0078	0.0063	18	10.00	4.30
0.84	7.00	0.0142	0.0070	18	18.00	5.50
0.92	7.00	0.0243	0.0120	22	27.97	8.35
1.00	7.00	0.0093	0.0063	26	10.00	4.96
1.00	7.00	0.0187	0.0093	26	20.00	7.26
1.07	7.00	0.0140	0.0069	30	14.00	6.26
1.07	7.00	0.0241	0.0119	30	24.00	9.19
1.07	7.00	0.0302	0.0149	30	30.00	11.12

adequate and relatively even distribution of Pareto solutions for Problem *P2*. The resulting Pareto frontier is shown in Figure 2.7(a).

Using the following values for the Δt and Δr parameters, the resulting set of candidate family members is reduced to a Smart Pareto set of designs that consists of the product family members: $\Delta t_V = 3$ m^3, $\Delta r_V = 7.5$ m^3, $\Delta t_P = 3$ MPa, $\Delta r_P = 7.5$ MPa, $\Delta t_\Gamma = \$700$, and $\Delta r_\Gamma = \$1500$. Figure 2.7(b) shows the designs included in the product family. Note that Figure 2.7(b) retains the overall shape of the design space shown in Figure 2.7(a) with a more sparse distribution of Pareto solutions. The Smart Pareto solutions are summarized in Table 2.1. The first four columns show the design variable values, and the final three columns show the functional objectives values. Each row is a design. At this point, the common platform has not yet been established. For this reason, the Smart Pareto set is only an approximation of the the product family. As discussed earlier, we can calculate the family fitness both before and after the establishment of the common product platform. In this way, we can estimate the performance cost to implement the family. The family fitness value for the pressure vessels prior to establishing a common platform is 1.5041.

Given thresholds of variation as follows: $\delta_R = 0.3$ m, $\delta_L = 1.2$ m, $\delta_{T_s} = 0.005$ m, and $\delta_{T_h} = 0.005$ m it is concluded, by comparing the maximum difference in each de-



(a) (b) (c)

Figure 2.7: (a) Pareto frontier, candidate family members (b) Smart Pareto set, product family members without a common platform (c) Smart Pareto set without a common platform (o) verses product family members sharing a common platform (x)

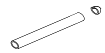
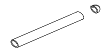
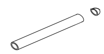
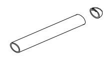
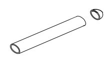
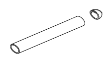
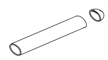
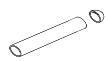
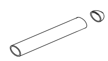
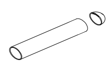
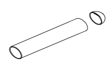
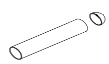
				KEY DIMENSIONS			
Prod.	V	P	Γ	R	L	T_s	T_h
	11.49	10.00	3.73	0.68	7.00	0.006	0.006
	10.00	18.00	4.27	0.64	7.00	0.011	0.006
	10.00	26.00	5.04	0.64	7.00	0.016	0.008
	18.00	10.00	4.30	0.84	7.00	0.008	0.006
	18.00	18.00	5.50	0.84	7.00	0.014	0.007
	16.00	27.95	6.79	0.80	7.00	0.021	0.010
	26.00	10.00	4.96	1.00	7.00	0.009	0.006
	26.00	20.00	7.26	1.00	7.00	0.019	0.009
	22.00	27.97	8.35	0.92	7.00	0.024	0.012
	30.00	14.00	6.26	1.07	7.00	0.014	0.007
	30.00	24.00	9.19	1.07	7.00	0.024	0.012
	30.00	30.00	11.12	1.07	7.00	0.030	0.015

Figure 2.8: Pressure vessel product family members with platform

sign variable, across all products in the family, to its respective threshold of variation, that length, L, is the only design variable that is adequately suited as a platform variable. The platform variable value of L is set to 7.0 m because, in this case, selecting the maximum maintains the feasibility of all products in the family and minimizes change as compared to the individual optimums. The product family sharing L as common platform is shown in Figure 2.8.

To summarize the change in performance resulting from the implementation of the product family, Figure 2.7(c) shows a plot of both the Smart Pareto set of products and the product family members sharing a common platform. By comparing the values shown in Table 2.1 and Figure 2.8 note that the performance of only one product changed. Incidently, the change in performance resulted in the shift of the design to another Pareto optimal solution. Note that according to the parameters used in the Smart Pareto filter, the resultant design is still significantly different from the other products in the family and is therefore included in the final set of family members. The calculated family fitness value for the family built on a platform of common length is 1.7984.

Alternative Case:

Because each radius requires a different die for the manufacture of each head, selecting R and T_h as the scalable variables is not very logical and would likely be cost prohibitive. The selection of poorly suited variables as platform variables is not altogether uncommon in product family optimization, as was illustrated by Simpson's universal electric motor example where radius was chosen as a platform variable rather than the more easily scaled stack length [6]. As an alternative solution to the pressure vessel platform previously discussed, a set of four radii values have been selected as platform values, by comparing δ_R to discrete intervals of products. The resultant changes in performance are summarized in Figure 2.9. Figure 2.10 compares the product family built on the set of platforms values to the original Smart Pareto set of products. The product family built on four radii platforms closely matches the Smart Pareto set. This product family could also be manufactured at lower cost because only four dies would be required. For the family built on a platform of four radii, the family fitness value is 1.7763.

By comparing the family fitness values of the two product family formulations, the difference in the performance can be seen. Because the optimization minimized each objective, a decrease in the family fitness value indicates an improvement in the average performance of the product family. The product family built on a common length has a higher fitness value than the family built on four radii values. Further, recall that the family with radii as its platform would be less costly to manufacture and should therefore be considered in the formulation of the product family.

2.6.2 Universal Electric Motor Example Problem

The electric motor example problem included in this section is a tri-objective problem based on the well-known bi-objective problem [33]. Previous work has solved this problem for a specified number of product family members each with a predetermined

				KEY DIMENSIONS			
Prod.	V	P	Γ	R	L	T_s	T_h
	10.00	10.00	3.66	0.68	5.98	0.006	0.006
	11.49	16.87	4.39	0.68	7.00	0.011	0.006
	11.49	24.36	5.20	0.68	7.00	0.016	0.008
	18.00	10.00	4.30	0.84	7.00	0.008	0.006
	18.00	18.00	5.50	0.84	7.00	0.014	0.007
	18.00	26.45	7.02	0.84	7.00	0.021	0.010
	26.00	10.00	4.96	1.00	7.00	0.009	0.006
	26.00	20.00	7.26	1.00	7.00	0.019	0.009
	26.00	25.88	8.83	1.00	7.00	0.024	0.012
	30.00	14.00	6.26	1.07	7.00	0.014	0.007
	30.00	24.00	9.19	1.07	7.00	0.024	0.012
	30.00	30.00	11.12	1.07	7.00	0.030	0.015

Figure 2.9: Pressure vessel product family members with four platforms

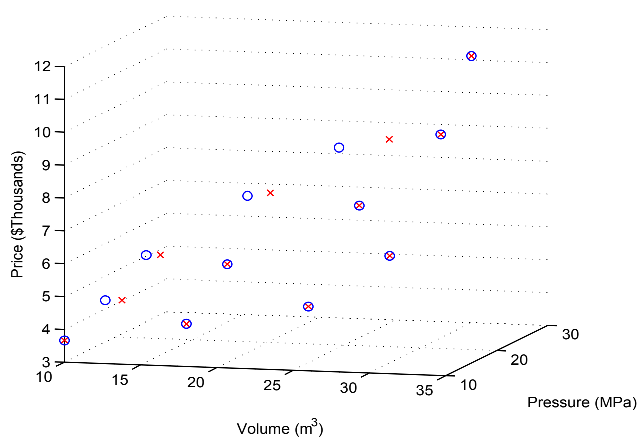


Figure 2.10: Product family of pressure vessels built on four radii platforms (x) verses Smart Pareto set (o)

torque constraint, as defined through market segmentation. In this formulation, torque is included as an objective and Smart Pareto filtering is used to determine the number of members to include in the product family as well as identify the members themselves. Note that two distinct sets of filter parameters are used with the Smart filter resulting in two different product families. This example illustrates the Smart filter's capability of leveraging designer knowledge of the product area and/or customer interests to define a product family. The problem statement is detailed in Problem 3.

Problem 3 (P3): Multiobjective Optimization of a Family of Electric Motors

$$\min_{x}[-T(x) \quad M(x) \quad -\eta(x)]^T \tag{2.24}$$

subject to:

$$t - r < 0 \tag{2.25}$$

$$H - 5000 \text{ Ampere/m} < 0 \tag{2.26}$$

$$0.15 - \eta \leq 0 \tag{2.27}$$

$$M - 2.0 \text{ Kg} \leq 0 \tag{2.28}$$

$$0.05 \text{ Nm} - T \leq 0 \tag{2.29}$$

$$T - 0.5 \text{ Nm} \leq 0 \tag{2.30}$$

$$P_o - 300 \text{ W} = 0 \tag{2.31}$$

where:

$$x = [N_a, N_f, t, I, L_s, A_a, A_f, r] \tag{2.32}$$

$$\eta = P_o/P_i \tag{2.33}$$

$$M = M_s + M_a + M_w \tag{2.34}$$

$$T = K\phi I \tag{2.35}$$

Table 2.2: Electric motor product family members before platform is selected, using first set of filter parameters

N_a	N_f	t	I	L_s	A_a	A_f	r	*Torque*	*Mass*	η
1156	54	6.88	5.98	26.16	0.25	0.25	25.60	0.5	0.763	0.436
1149	51	6.52	5.95	24.48	0.24	0.24	24.34	0.422	0.656	0.439
1065	49	5.70	5.83	23.74	0.21	0.21	22.67	0.33	0.534	0.447
1056	50	5.30	5.35	21.69	0.19	0.19	21.20	0.242	0.436	0.488
723	64	4.61	4.24	25.85	0.17	0.17	21.48	0.164	0.429	0.615
726	66	4.04	3.16	17.60	0.21	0.21	16.58	0.05	0.246	0.826

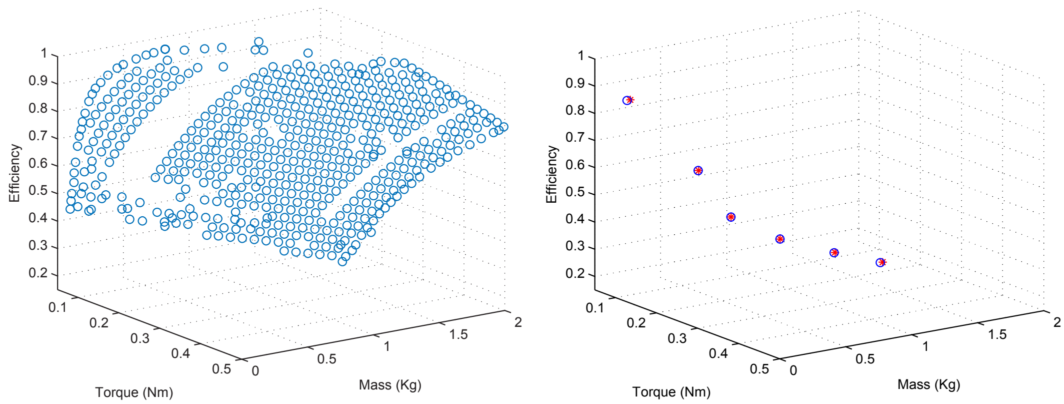
Additional supporting equations were used in the derivation of the objective functions as detailed within the problem testbed [33].

P3 was solved using the Normal Constraint method [32], thereby generating a discrete representation of the Pareto frontier. The modified smart Pareto filter was used to preferentially select products of lower mass and higher efficiency; filter parameters were defined as follows:

(1)$\Delta\eta \leq 0.05$ is insignificant, an increase $\Delta\eta > 0.12$ is notable, and no feasible decrease in η is considered noteworthy.
(2)$\Delta M \leq 0.25$ Kg is insignificant, a decrease $\Delta M > 0.75$ Kg is notable, and no feasible increase in mass is considered noteworthy.
(3) $\Delta T \leq 0.03$ Nm is insignificant and $\Delta T > 0.075$ Nm is noteworthy.

The filter determined the number of members to include in the product family and identified the members themselves. The resultant set of product family members is summarized in Table 2.2. Family fitness was calculated according to Equation 2.8, resulting in a fitness value of 1.5518. Figure 2.11 shows the discrete representation of the Pareto frontier and a comparison plot of the Smart Pareto set verses the product family, showing the performance changes resulting from the establishment of the platform. Note that the resulting Smart Pareto set is a curve rather than a surface as might be expected. This is the result of the designer's preference for low mass and high efficiency as specified by conditions 1 and

2 above. Careful examination of these conditions shows that they cause large sets of points to be removed from the efficiency-mass plane, at various intervals of torque.



(a) (b)

Figure 2.11: (a)Pareto frontier for universal electric motor, (b) Smart Pareto set (o) verses product family (*) using first set of filter parameters

Using a second set of filter parameters, as follows:

(1)$\Delta\eta \leq 0.05$ is insignificant, an increase $\Delta\eta > 0.12$ is notable, and no feasible decrease in η is considered noteworthy.

(2)$\Delta M \leq 0.25$ Kg is insignificant, a decrease $\Delta M > 0.50$ Kg is notable, and no feasible increase in mass is considered noteworthy.

(3) $\Delta T \leq 0.03$ Nm is insignificant and $\Delta T > 0.04$ Nm is noteworthy.

A second product set was identified that more closely matches the benchmark set [33]. This set of product family members, family fitness of 1.5442, is summarized in Table 2.3. Figure 2.12 shows the Smart Pareto set of products as well as the product family sharing a common platform.

It is important to note that the two different sets of filter parameters used in this example result in a varied number of product family members. Notably, the product set identified by the first set of filter parameters consists of fewer products that span the same design space and have more performance distinctiveness.

Table 2.3: Electric motor product family members before platform is selected, using second set of filter parameters

N_a	N_f	t	I	L_s	A_a	A_f	r	$Torque$	$Mass$	η
1156	54	6.88	5.98	26.16	0.25	0.25	25.60	0.5	0.763	0.436
1162	54	6.74	5.86	25.46	0.25	0.25	25.03	0.46	0.719	0.445
1149	51	6.47	6.00	23.45	0.24	0.19	24.20	0.403	0.624	0.435
1072	50	5.86	5.79	24.20	0.22	0.22	23.15	0.349	0.566	0.451
1041	48	5.81	5.93	21.32	0.20	0.20	22.62	0.292	0.471	0.44
1057	63	6.77	4.80	19.90	0.20	0.21	24.13	0.248	0.504	0.544
1044	66	5.20	4.02	25.07	0.23	0.23	21.02	0.205	0.53	0.649
723	64	4.61	4.24	25.85	0.17	0.17	21.48	0.164	0.429	0.615
668	67	3.95	3.55	23.55	0.18	0.18	18.90	0.091	0.329	0.736
726	66	4.04	3.16	17.60	0.21	0.21	16.58	0.05	0.246	0.826

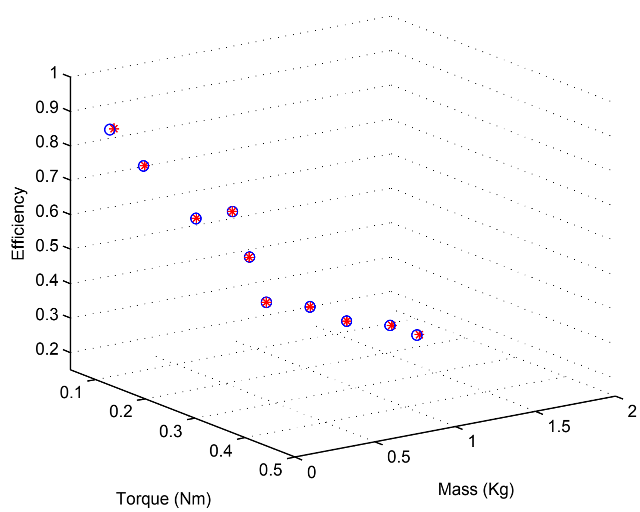


Figure 2.12: Smart Pareto set (o) verses product family (*) using second set of filter parameters

For each of these product sets a common platform was determined. Figures 2.11 and 2.12 show the change in performance that resulted from the implementation of each product platform. The resulting product families are summarized in Tables 2.4 and 2.5. The family fitness values using the first and second sets of designer preferences, 1.5579 and 1.5484 respectively, are slightly higher (decrease average performance for a minimiza-

Table 2.4: Electric motor product family members after platform is selected, using first set of filter parameters

N_a	N_f	t	I	L_s	A_a	A_f	r	$Torque$	$Mass$	η
1156	46	5.51	5.98	34.31	0.27	0.27	21.98	0.5	0.78	0.436
1149	46	5.51	5.95	29.25	0.25	0.20	21.98	0.422	0.663	0.439
1065	47	5.51	5.83	25.15	0.21	0.21	21.98	0.33	0.535	0.447
1056	52	5.51	5.35	20.28	0.19	0.19	21.98	0.242	0.437	0.488
723	65	5.51	4.24	25.36	0.17	0.17	21.98	0.164	0.432	0.615
725	88	5.51	3.16	10.34	0.23	0.23	21.98	0.05	0.27	0.826

Table 2.5: Electric motor product family members after platform is selected, using second set of filter parameters

N_a	N_f	t	I	L_s	A_a	A_f	r	$Torque$	$Mass$	η
1156	47	5.63	5.98	33.53	0.26	0.26	22.27	0.5	0.777	0.436
1162	48	5.63	5.86	31.30	0.26	0.22	22.27	0.46	0.73	0.445
1149	47	5.63	6.00	27.07	0.24	0.22	22.27	0.403	0.626	0.435
1072	48	5.63	5.79	26.03	0.22	0.21	22.27	0.349	0.567	0.451
1041	47	5.63	5.93	21.89	0.20	0.20	22.27	0.292	0.471	0.440
1057	58	5.63	4.80	22.61	0.21	0.21	22.27	0.248	0.501	0.544
1044	70	5.63	4.02	22.59	0.23	0.23	22.27	0.205	0.529	0.649
723	66	5.63	4.24	24.78	0.17	0.17	22.27	0.164	0.433	0.615
669	79	5.63	3.55	17.81	0.17	0.18	22.27	0.091	0.337	0.736
725	89	5.63	3.16	10.11	0.23	0.23	22.27	0.05	0.272	0.826

tion problem) for each product family due to the performance changes that result from the implementation of the product platform.

In comparing the results from the second set of designer-preferences to the benchmark family, it is noted that the family fitness of the product family from this example is slightly higher than that of the benchmark family, 1.5442 compared to 1.5260. The higher family fitness of the family from this example indicates a decrease in objective performance of 1.2 percent.

2.7 Concluding Remarks

This paper presented a method of product family design that can be used to determine the number of members to include in a product family, identify the members themselves, and define the product platform. The method is of particular use when traditional market segmentation data is unavailable or is not easily gathered or distilled by the company designing the product family. According to the methodology, (i) a discrete representation of a Pareto frontier comprised of candidate family members is identified. (ii) A filter is then used to identify a Smart Pareto set of products that constitutes the product family. This Smart Pareto set includes the members needed to make a product family sufficiently diverse and represents the least number of products that provide adequate representation of product performance tradeoffs. (iii) A designer-specified maximum allowable difference in each design variable, threshold of variation, is used to determine which of the design variables are best suited as platform variables and which are best suited as scalable variables. (iv) Finally, the product family, sharing a common platform, is compared to the set of individually optimized products to show the performance change that results from the implementation of the product family. The result is a minimal set of products, designed on a common platform, that together comprise a product family.

The approach presented herein was demonstrated with two popular benchmark problems; the design of a family of pressure vessels and the design of a family of electric motors. In both cases, the design framework resulted in a conceivable and reasonable family of products.

We note that in some cases, product families will require highly varying spaces between products in the product family along one or more objective axes. Such variability may be required when tightly grouped subsets of products can be used, for example, to discourage competitors from entering a particular market segment. The filtering method presented in this paper does not provide a means for the designer to specify filter parameters as a function of position along an objective axis and therefore does not generally result in highly varying spaces between family members, although it can happen naturally depending on the objective functions scales and the nature of the Pareto frontier. This type of flexibility in specifying filter parameter would allow the filter to identify product families with highly varying spaces between products in that family. The development of this flexibility is the focus of our next phase of research in this area.

Chapter 3

Product Family Member and Platform Identification with Concurrent Variable and Objective Space Smart Pareto Filtering

3.1 Abstract

Product families are frequently used to provide consumers with a variety of appealing products and to help maintain reasonably low production costs for manufacturers. Three common objectives in the design of product families are used to balance the interests of both consumers and manufacturers. These objectives are to maximize product (i) performance, (ii) distinctiveness as perceived by the consumer, and (iii) commonality as perceived by the manufacturer. To accomplish these objectives, two general approaches to product family design are frequently implemented: scale-based and module-based design. Important to both of these design approaches is the selection of product family members and the identification of common features that can be shared by family members. In this paper, a Pareto filter that concurrently considers both objective and variable spaces is used to identify a product family where each product in the family possesses optimal and distinct performance characteristics. Simultaneously, the filter also identifies features that can be shared by the products in the family. The filter functions as follows: (1) Given a set of Pareto solutions, a starting design is identified and classified as the first member of the product family. (2) A region of insignificant tradeoff is constructed about the starting design in objective space. Any design located within the region of insignificant tradeoff is considered insignificantly different from the starting design and is eliminated from the set of candidate product family members. (3) Each remaining candidate product family member is ranked according to the effect that its addition to the product family would have on the variation among design variables. Specifically, the design resulting in the lowest calculated standard

deviation to mean ratio, summed across all design variables, is selected as the next member of the product family. (4) A region of insignificant tradeoff is constructed about this design and any designs located within this space are removed from the set of candidate product family members. Steps 3 and 4 are then repeated until all points have either been included as a member of the product family or identified as being insignificantly different from at least one product family member. Considering all products in the family, if for any design variable the variation from product to product in the family is sufficiently small, then that variable is identified as well suited to become common to the entire family. A well-known universal electric motors problem is used to demonstrate the method.

3.2 Nomenclature

x	Vector of design variables
n_x	Number of design variables
μ	Vector of design metrics (objectives)
n_μ	Number of design metrics (objectives)
g	Vector of inequality constraints
h	Vector of equality constraints
n_φ	Number of candidate product family members (solutions on discrete Pareto frontier)
$\Delta t, \Delta r$	Filter parameters defining a region of practically insignificant objective-tradeoff
Δp^i	Summation of the squared differences between each variable in product p^i and the mean of the same variable across all candidate family members
s^i	Summation of the standard deviation to mean ratios for all variables in product p^i
$\bar{x}_j^j$	Mean of design variable x_j for the product set including Smart Pareto designs and product p^i
n_ξ	Number of products in product family
ξ	Matrix of product family members prior to the establishment of the product platform
ψ	Matrix of product family members after the establishment of the product platform
J_{p^i}	Aggregate objective function for product p^i
F	Average fitness of product family

ΔF_{n_v}	Normalized change in family fitness per platform variable
n_v	Number of platform variables
N_a	Number of wire turns on armature
N_f	Number of wire turns on field, per pole
t	Thickness of stator
I	Electric current
L_s	Stack length
A_a	Armature wire cross-sectional area
A_f	Field wire cross-sectional area
r	Outer radius of stator
η	Efficiency
M	Motor mass
T	Motor torque
H	Magnetizing intensity
P_o	Gross mechanical power output
P_i	Electrical input power
M_s	Stator mass
M_a	Armature mass
M_w	Windings mass
K	Motor constant
ϕ	Magnetic flux

3.3 Introduction and Literature Survey

In today's world marketplace, consumers increasingly demand variety, customization, and personalization in the goods they purchase. For manufacturers, economically satisfying these demands is a notable challenge. In many instances, however, product families provide consumers a variety of appealing products and help manufacturers maintain reasonably low production costs through increases in economies of scale and scope [3]. Three goals, common in the design of product families, are considered when balancing the interests of both consumers and manufacturers. These goals are to maximize product (i) performance, (ii) distinctiveness, and (iii) commonality [4, 5]. In the literature, a number of design and optimization approaches exist that address the three goals of product family design [6]. Many of these approaches identify a set of design features that are shared by product family members. In the literature, this set of shared features is known as a product family platform [34].

Traditionally, product families have been built on two primary platform types: module-based platforms and scale-based platforms [35]. A module-based platform is the foundation for a collection of related products that have differing functions through the addition or subtraction of modules. A scale-based platform is the foundation of related products that have differing function through the scaling of non-platform design features. Essential to product families built on either type of platform are the selection of product family members and the identification of product platform features.

Many of the approaches included in the literature use a heuristic tool, market segmentation, to identify product family members [36]. The set of product family members thus identified is subsequently optimized to improve product performance and to identify a product platform. These approaches produce encouraging results but may introduce sub-optimality because product family member identification is completed independent of platform selection. Improved product family optimality, in many cases, can be achieved when the selection of product family members and the identification of product platform features are completed simultaneously.

Additionally, when sufficient market data is unavailable or is not easily obtained or distilled by the company developing the product family, traditional segmentation methods become difficult to implement. For example, entrepreneurs and small to mid-sized companies may not have a dedicated marketing department or the financial resources needed to purchase market research from an outside firm. When planning to launch a series of products built on a common platform, these small companies may struggle to gather and process the market data needed to construct the segmentation grid traditionally used to identify product family members. As an alternative to traditional segmentation, the authors previously introduced the use of Smart Pareto filters as a method for selecting family members [23]. Thus used, Smart Pareto filtering leverages the designer's knowledge of the product area and/or customer interests when selecting members of a product family.

In this paper, multiobjective optimization is used to identify a set of candidate product family members and concurrent Smart Pareto filtering in objective and variable spaces is used to simultaneously select product family members and identify the product family platform. The method requires that a meaningful multiobjective optimization problem be formulated and solved for a discrete representation of a Pareto frontier. A Smart Pareto filter is then used to eliminate overly similar designs and thereby select product family members from among the Pareto solutions.

The filtering of the Pareto frontier is important to product family member selection because, as Mattson et al. [28] observe, many designs, corresponding to solutions within the discrete representation of the Pareto frontier, are practically indistinguishable from one another. Further, Nelson et al. [31] note that the effectiveness of the product family decreases as the performance characteristics of the products within the family become less distinct. Traditional Smart Pareto filters [28], including the filter previously used by the authors for family member selection [23], function in objective space only. However, the concurrent Smart Pareto filter presented herein differs from previous filters in that in addition to searching in objective space, it seeks to minimize variable space variations during the selection of the product family members, thereby improving the ease with which the product platform is established.

The filter functions as follows: (1) Given a set of Pareto solutions, a starting design is identified and classified as the first member of the product family. (2) A region of practically insignificant objective-tradeoff is constructed about the starting design in objective space. Any design located within the region of insignificant tradeoff is determined to be indistinguishable from the starting design and is removed from the set of candidate product family members. (3) Each remaining design is considered as a subsequent member of the product family and is ranked according to the effect that the addition of each would have on the variation among the design variables. Specifically, the design causing the smallest increase in the calculated standard deviation to mean ratio, summed across all design variables, is selected as the next family member. (4) A vector space is then constructed about the subsequent design and any additional designs located within region of insignificant tradeoff are again removed from the set of candidate family members. Steps 3 and 4 are repeated until all points have either been included in or deemed unimportant to the product family.

The remainder of this paper is presented as follows. A preliminary review of a generic multiobjective optimization problem is included in Section 3.4. In Section 3.5, concurrent Smart Pareto filtering in variable and objective spaces is presented in its theoretical form. A well-know electric motor example illustrating the application of the method is included in Section 3.6, with our concluding remarks given in Section 3.7.

3.4 Technical Preliminary: Multiobjective Optimization

The use of the any Pareto filtering method, including concurrent Smart Pareto filtering in variable and objective spaces, requires that a meaningful multiobjective optimization problem be formulated and solved to obtain a Pareto set. A generic, mathematical description of a multiobjective optimization problem is included in Problem 1 for completeness.

Problem 1 (P1): Generic Multiobjective Optimization

$$\min_{x} \mu(x) \tag{3.1}$$

subject to:

$$g(x) \leq 0 \tag{3.2}$$

$$h(x) = 0 \tag{3.3}$$

$$x_l \leq x \leq x_u \tag{3.4}$$

When solved, multiobjective optimization problems, such as *P1*, result in many Pareto solutions that together comprise a discrete representation of the Pareto frontier. Figure 3.1 shows a hypothetical Pareto frontier for a bi-objective minimization problem. The shaded region in this figure represents feasible space, wherein all solutions satisfy every

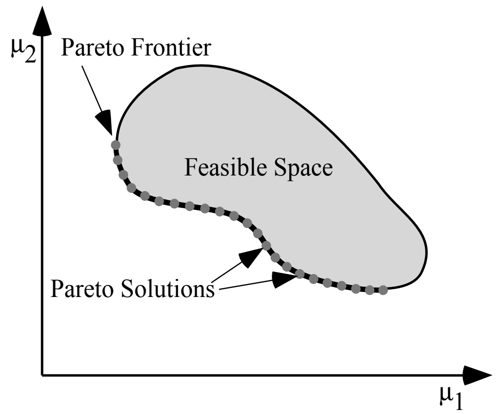


Figure 3.1: Discrete representation of Pareto frontier

constraint. However, because both objectives in this problem are minimized, the solutions found on the lower left hand boundary of the feasible space, the Pareto solutions, are of particular interest. Importantly, each Pareto solution is non-dominated, meaning that, for each solution, no other solution exists that possesses equal or improved performance in every objective. Also, each solution is mathematically optimal, meaning that, for a single solution, any improvement in an objective can only occur at the expense of another objective [32]. Because of these important characteristics of Pareto solutions, the discrete set of Pareto solutions captures objective tradeoffs between conflicting design objectives and consists of many functionally different design alternatives.

3.5 Concurrent Smart Pareto Filtering in Variable and Objective Spaces

Because it consists of many optimal yet functionally different design alternatives, a Pareto frontier can be considered a set of candidate product family members. Further, through filtering, a Pareto frontier can be leveraged to directly satisfy two of the goals of product family design: maximum product performance and distinctiveness. The third goal of product family design, maximum commonality, can also be satisfied by using a filter to select designs, from among Pareto solutions, that are similar with respect to certain design variables. Considering both objective and variable spaces, concurrent Smart Pareto filtering is used to address the three goals of product family design simultaneously by selecting product family members with optimal and distinct performance as well as by identifying shared platform variables. The simultaneous selection of family members and platform variables increases the potential of designing a product family with high levels of product commonality and only minimal performance tradeoffs.

The process of reducing the Pareto frontier to a product family is an iterative process. The flowchart of Figure 3.2 illustrates this process. Each of the primary steps of the process are detailed in the following sections: In Section 3.5.1 a method for selecting a starting design is presented. In Section 3.5.2, the process of removing designs that are insignificantly different from other product family members is described. Section 3.5.3 then details the selection of subsequent designs according to the variation that would be introduced through the addition of a specific design. Section 3.5.4 details the calculation of the average objective performance of the product family or *family fitness*. Additionally, the normalized change in family fitness per platform variable is calculated as a measure of the effectiveness of the product family platform.

3.5.1 Selecting a Starting Design (Single Pareto Point)

The first step of any Smart Pareto filtering process is the selection of a starting design. Traditional Smart filters typically select one of the anchor points of the Pareto frontier, a Pareto solution possessing the best possible value in a single objective, as their starting design. While the Smart set subsequently identified by the traditional Smart filter varies slightly with the selected starting design, when filtering in objective space only, the differences in the Smart sets that result from using the different anchor points as the starting design are generally small. In most cases, traditional Smart filters will produce acceptable results regardless of which anchor point is used as the starting design.

In contrast, when using concurrent Smart Pareto filtering, the selection of an anchor point as the starting design will generally yield suboptimal results. This is the case because when using concurrent filtering, the starting design directly impacts decisions made in objective *and* variable spaces. Ultimately, the poor selection of a starting design complicates the establishment of the product family platform. Therefore, when selecting a starting de-

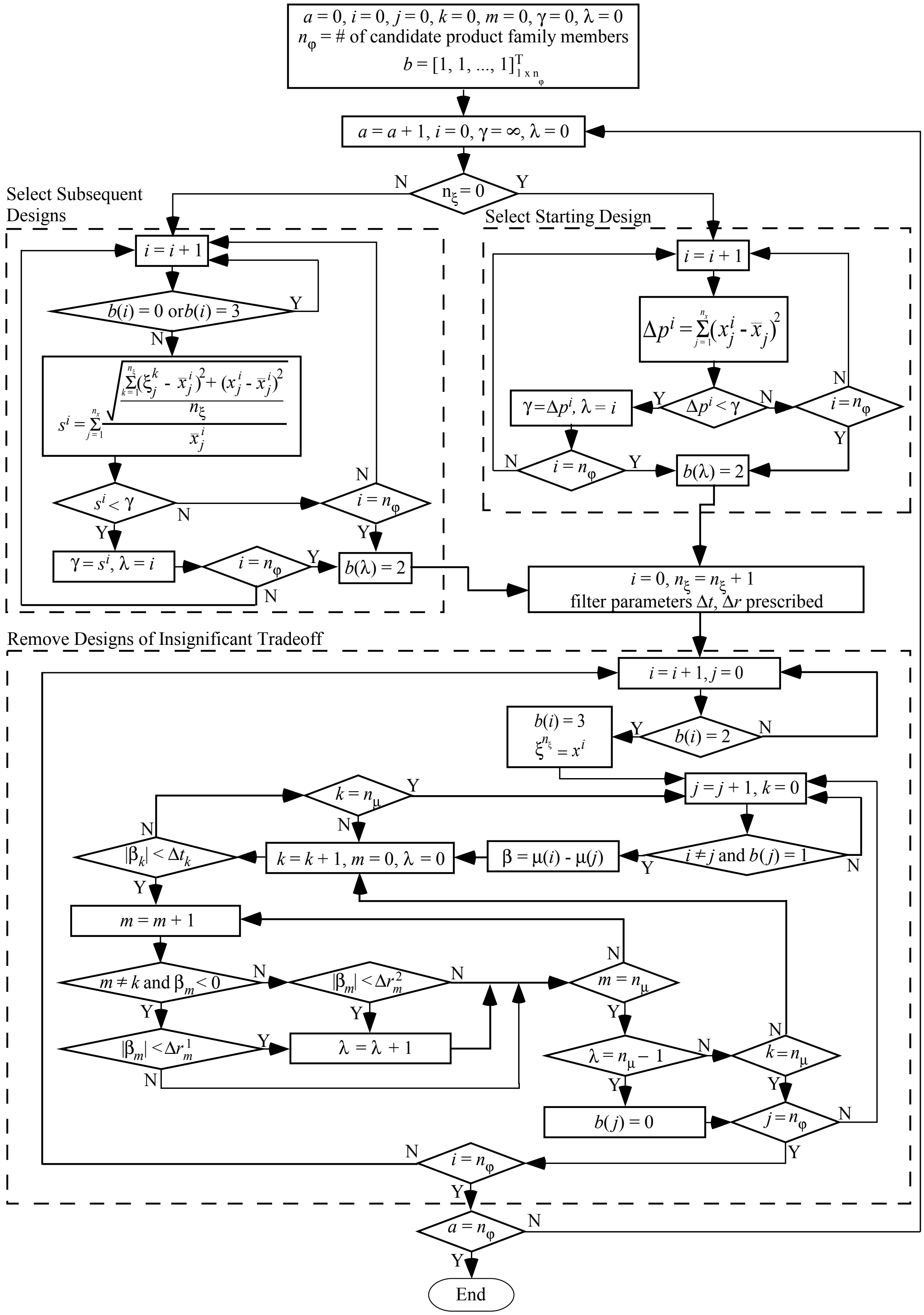


Figure 3.2: Algorithmic flowchart of concurrent Smart Pareto filtering in variable and objective spaces

sign for use with concurrent filtering, it is desired that the starting design be similar, at least in variable space, to the many candidate product family members. Using a design where each variable deviates minimally from the central tendency of the same variable, across all Pareto solutions, improved filtering results are obtained. Specifically, the summation of the squared differences between each variable and that variable's respective mean is used to rank each candidate starting design. The design with the smallest sum is selected as the starting design and becomes the first member of the product family. Equation 3.5 details this calculation.

$$\Delta p^i = \sum_{j=1}^{n_x} (x_j^i - \bar{x}_j)^2 \tag{3.5}$$

The process of selecting a starting design is detailed in the upper right-hand section of Figure 3.2. The three steps in this portion of the algorithm are described as follows: (1) Initialize the algorithm indices and variables: a = 0, i = 0, $\gamma = \infty$, λ = 0, and b = index matrix of size (1 x n_φ). (2) For each Pareto solution perform the calculation detailed in Equation 3.5. Store in the index variable λ the location of the design resulting in the lowest total summation. (3) Update the index matrix at location λ so that $b(\lambda) = 2$. The filtering process then continues, as described in Section 3.5.2, by removing any designs that are insignificantly different, in terms of objective performance, from the design located at λ.

3.5.2 Removing Designs of Insignificant Objective-Tradeoff

Having selected a starting design, the second step of the filtering process is to remove any designs that are insignificantly different, in terms of objective performance, from the starting design. To accomplish this, a Smart Pareto filter, defining a *Region of Insignificant Tradeoff*, is constructed and centered on the starting design, labeled $\hat{\mu}$, as shown in Figure 3.3. Pairwise comparisons are then made between the starting design and all other candidate product family members. When these comparisons identify a design that is located within the region of insignificant tradeoff, the design is removed from the set of candidate family members.

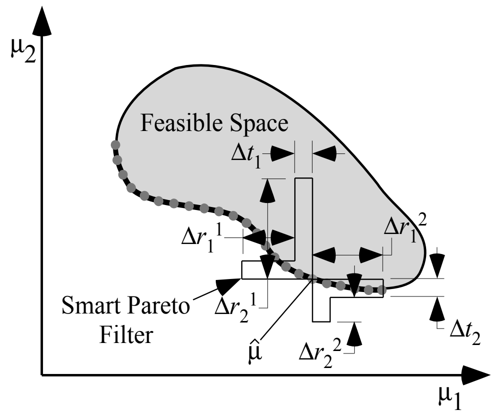


Figure 3.3: 2-D Smart Pareto filter

The geometric shape of the region of insignificant tradeoff is defined by a set of designer specified parameters, $\Delta t = \Delta t_1, \Delta t_2, ..., \Delta t_{n_\mu}$ and $\Delta r = \Delta r_1, \Delta r_2, ..., \Delta r_{n_\mu}$, where $\Delta r_i = \Delta r_i^1, \Delta r_i^2, ..., \Delta r_i^{i-1}, \Delta r_i^{i+1}, ..., \Delta r_1^{n_\mu}$, for $i = 1...n_\mu$. Figure 3.3 shows each of these parameters for the bi-objective case. The two parameters, Δt and Δr can be understood by the following; when comparing two Pareto points (or designs), any difference in μ_i that is less than Δt_i is considered to be insignificant. As such, one of the two designs being compared should be removed since it is insignificantly different – at least in one objective. When, however, the difference in μ_i, between two points, is within Δt_i, but the change in another objective (μ_j) is significantly large (greater than Δr_j), then the design should not be removed. In other words, any change in μ_j that is larger than Δr_j is deemed significant – regardless of how small the change in another objective is. In this way, the parameters Δt and Δr are physically meaningful and can be easily specified by the designer.

The comparison process used to identify designs that are insignificantly different from product family members is detailed in the lower section of Figure 3.2. The six steps in this portion of the algorithm are described as follows: (1) Initialize the algorithm indices and variables: $i = 0$, $j = 0$, $k = 0$, $m = 0$, and $\lambda = 0$. (2) Locate the most recently identified product family member as indicated by an index matrix value of $b(i) = 2$. Update the index matrix at that location so that $b(i) = 3$. (3) Define the vector β by subtracting the design metrics (objectives) of each remaining candidate family member from the design metrics of the recently identified member of the product family. (4) Compare the vector element β_k to the filter parameter Δt_k. If for any element the absolute value of β_k is less than Δt_k, proceed to step 5. Otherwise, the design in question is determined to be sufficiently different from the recently identified family member and, as a result, remains within the set of candidate designs. (5) Compare all other elements of the vector β to the corresponding filter parameter Δr_m. If for all vector elements, other than β_k, the absolute value of β_m is less than Δr_m, proceed to step 6. Otherwise, the design in question is determined to be sufficiently different from the recently identified family member and, as a result, remains within the set of candidate designs. (6) For all designs located within the region of insignificant tradeoff, as determined by steps 4 and 5 collectively, update the index matrix at the corresponding location so that $b(j) = 0$. Importantly, as noted in Section 3.5.3, each design where $b(j) = 0$ is removed from the set of candidate designs and is no longer considered during the selection of subsequent product family members.

3.5.3 Selecting Subsequent Smart Set Designs

As noted previously, product family design has as its goal the maximization of product performance, distinctiveness, and commonality. Because product family members are selected from the set of Pareto solutions and insignificantly different designs are removed from the set of candidate product family members, the designs selected as members of the product family are, in terms of functional performance, optimal and distinct. The third goal of product family design, maximum commonality, is more easily achieved if the designs subsequently chosen as product family members introduce only small variation in design variable space.

Each subsequent design is selected according to the impact that its addition to the product family has on the summation of standard deviation to mean ratios for all design variables and all included family members. Equation 3.6 summarizes the calculation used to rank each design.

$$s^i = \sum_{j=1}^{n_x} \frac{[(\sum_{k=1}^{n_\xi}(\xi_j^k - \bar{x}_j^i)^2 + (x_j^i - \bar{x}_j^i)^2)/n_\xi]^{1/2}}{\bar{x}_j^i} \tag{3.6}$$

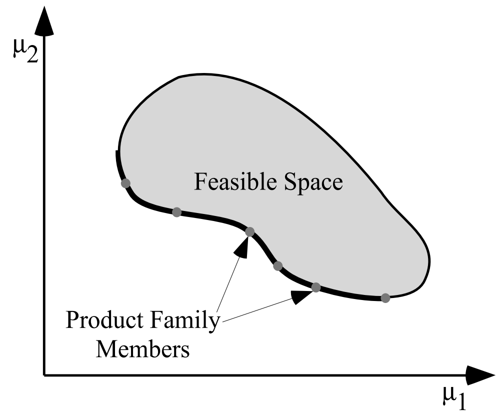


Figure 3.4: Product family resulting from concurrent Smart Pareto filtering

The process of selecting subsequent designs is detailed in the upper left-hand section of Figure 3.2. The three steps in this portion of the algorithm are described as follows: (1) Initialize the algorithm indices and variables: $i = 0$, $\gamma = \infty$, and $\lambda = 0$. (2) For each Pareto solution that has not been removed from the set of candidate family members, where removed designs are indicated by index matrix values of 0 or 3, perform the calculation detailed in Equation 3.6. Store in the index variable λ the location of the design resulting in the lowest total summation. (3) Update the index matrix at location λ so that $b(\lambda) = 2$. The filtering process then continues, as described in Section 3.5.2, by removing any designs that are insignificantly different, in terms of their objective performance, from the design located at λ.

After a subsequent design is selected, the process described in Section 3.5.2 is then repeated with the filter centered on the design resulting in the lowest total summation, as calculated using Equation 3.6. These steps are iteratively continued until all product family members are selected and all other designs are identified as being insignificantly different from at least one product family member. Considering all products in the family, if for any design variable the variation from product to product in the family is sufficiently small, then that variable becomes a platform variable and is made common to the entire family. Figure 3.4 shows the hypothetical product family that results from the process.

3.5.4 Measuring the Average Performance of the Product Family

In addition to the increase in commonality that results from the establishment of the product platform, a decrease in objective performance typically occurs. To evaluate the extent of these performance changes, the product family after the establishment of the platform is compared to the family prior to the platform. The objective performance in both cases is evaluated numerically by calculating a *family fitness* value. Family fitness is an average measure of objective performance across the entire product family and is calculated using a normalized aggregate objective function as shown in Equation 3.7.

$$F = \sum_{i=1}^{n_\xi} (J_{p^i})/n_\xi \tag{3.7}$$

In the case of a minimization problem, a smaller family fitness value corresponds to a higher average performance for the product family. Note, however, that family fitness exclusively considers the objective performance of the family as a whole and does not consider commonality in its calculation. Instead, commonality is quantified, for a product family built on a scale-based platform, simply as the number of design variables common to all products in the family, n_v.

Additionally, the performance decrease resulting from the establishment of the platform can be evaluated, in terms of the increase in commonality, by using the family fitness values from both before and after the platform implementation. Specifically, the normalized change in family fitness per platform variable is calculated as a measure of the effectiveness of the platform approach. Equation 3.8 details this calculation, where F^ξ and F^Ψ are respectively the family fitness values of the product family before and after the establishment of the platform.

$$\Delta F_{n_v} = (F^\Psi - F^\xi)/(F^\xi \cdot n_v) \tag{3.8}$$

In the section that follows, Equations 3.7 and 3.8 are used to compare the family of products identified using concurrent filtering to a benchmark family of universal electric motors [12].

3.6 Universal Electric Motor Example Problem

The example problem included in this section is a tri-objective problem based on the well-known bi-objective problem [33]. In this example, given a set of filter parameters, a traditional Smart Pareto filter that considers only objective space is used to select product family members. Then, using the same filter parameters, concurrent Smart Pareto filtering in variable and objective spaces is used to select product family members and to identify the design variables that are well suited to become platform variables. To show the effectiveness of the method, for each variable in the two resulting product families, the standard deviation to mean ratios are compared. The problem statement is detailed in Problem 2.

Problem 2 (P2): Multiobjective Optimization of a Family of Electric Motors

$$\min_{x}[-T(x) \quad M(x) \quad -\eta(x)]^T \tag{3.9}$$

subject to:

$$\begin{aligned} t-r &< 0 && (3.10)\\ H-5000\ \text{Ampere/m} &< 0 && (3.11)\\ 0.15-\eta &\leq 0 && (3.12)\\ M-2.0\ \text{Kg} &\leq 0 && (3.13)\\ 0.05\ \text{Nm}-T &\leq 0 && (3.14)\\ T-0.5\ \text{Nm} &\leq 0 && (3.15)\\ P_o-300\ \text{W} &= 0 && (3.16) \end{aligned}$$

where:

$$\begin{aligned} x &= [N_a, N_f, t, I, L_s, A_a, A_f, r] && (3.17)\\ \eta &= P_o/P_i && (3.18)\\ M &= M_s+M_a+M_w && (3.19)\\ T &= K\phi I && (3.20) \end{aligned}$$

Additional supporting equations were used in the derivation of the objective functions as detailed within the problem testbed [33].

P2 was, first, solved using a Normal Constraint [32] optimization approach. Then, using the set of filter parameters detailed below, the discrete representation of the resultant Pareto frontier was reduced to an individually optimized set of product family members.

(1)$\Delta\eta \leq 0.05$ is insignificant, an increase $\Delta\eta > 0.12$ is notable and no feasible decrease in η is considered noteworthy.
(2)$\Delta M \leq 0.25$ Kg is insignificant, a decrease $\Delta M > 0.50$ Kg is notable, and no feasible increase in mass is considered noteworthy.
(3) $\Delta T \leq 0.03$ Nm is insignificant and an increase or decrease $\Delta T > 0.04$ Nm is notable.

A traditional Smart Pareto filter, considering only objective space, was applied first, followed by concurrent Smart Pareto filtering in variable and objective spaces. The results of the two methods are included in Tables 3.1 and 3.2. Each row in Tables 3.1 and 3.2 corresponds to an individual product design within the respective product family. The percentages at the bottom of each table are the calculated standard deviation to mean ratios for each design variable. Both product families are plotted in objective space in Figure 3.5.

Comparing the standard deviation to mean ratio for each design variable, between the product families shown in Tables 3.1 and 3.2, it is observed that for six of the eight design variables the standard deviation to mean ratio was reduced by using concurrent filtering. Further, by calculating an average standard deviation to mean ratio for each of the product families, it is noted that the average ratio was reduced from approximately 15.9 percent when using a traditional filter to 10.1 percent by using concurrent filtering–a decrease of over 35 percent.

Perhaps most important, according to Nayak et al. [37], for this particular problem, design variables with a standard deviation to mean ratio of 10 percent or less are considered well suited to become platform variables. Following this guideline, filtering only in objective space failed to identify a single variable as a likely platform variable. Concurrent

Table 3.1: Product family using Smart Pareto filter in objective space only with standard deviation to mean ratios for each design variable

N_a	N_f	t	I	L_s	A_a	A_f	r	*Torque*	*Mass*	η
726	66	4.04	3.16	17.60	0.21	0.21	16.57	0.05	0.25	0.83
668	67	3.95	3.54	23.55	0.18	0.18	18.89	0.09	0.33	0.74
723	64	4.61	4.24	25.85	0.17	0.17	21.48	0.16	0.43	0.62
1044	66	5.20	4.02	25.07	0.23	0.23	21.02	0.20	0.53	0.65
1057	63	6.77	4.80	19.90	0.21	0.20	24.13	0.25	0.50	0.54
1041	48	5.81	5.93	21.32	0.20	0.20	22.62	0.29	0.47	0.44
1072	50	5.86	5.79	24.20	0.22	0.22	23.15	0.35	0.57	0.45
1149	51	6.47	6.00	23.45	0.19	0.24	24.20	0.40	0.62	0.43
1162	54	6.74	5.86	25.46	0.25	0.25	25.03	0.46	0.72	0.45
1156	54	6.88	5.98	26.16	0.25	0.25	25.59	0.50	0.76	0.44
19.9%	13.1%	20.0%	22.6%	12.1%	13.1%	13.3%	12.8%			

Table 3.2: Product family using concurrent Smart Pareto filtering in variable and objectives spaces with standard deviation to mean ratios for each design variable

N_a	N_f	t	I	L_s	A_a	A_f	r	*Torque*	*Mass*	η
676	89	6.59	2.99	18.42	0.31	0.32	21.27	0.07	0.42	0.87
672	97	7.00	3.14	21.86	0.29	0.29	24.33	0.11	0.56	0.83
675	93	5.84	3.31	30.36	0.29	0.29	24.52	0.18	0.73	0.79
1021	84	6.37	3.61	24.53	0.29	0.29	24.25	0.23	0.70	0.72
1051	86	7.26	3.74	25.73	0.29	0.29	25.65	0.28	0.79	0.70
1039	86	7.26	3.88	27.40	0.29	0.29	26.68	0.33	0.87	0.67
1070	86	7.57	4.01	28.24	0.30	0.29	27.50	0.38	0.95	0.65
1149	80	7.54	4.28	29.46	0.30	0.30	27.37	0.45	1.00	0.61
1154	78	7.59	4.47	30.18	0.30	0.30	27.78	0.50	1.04	0.58
22.0%	6.8%	8.7%	13.6%	15.5%	2.8%	3.4%	8.3%			

filtering, on the other hand, identified five of the eight variables as well suited to become platform variables.

A product platform was established with each of the five variables identified by concurrent filtering becoming common to all product family members. Table 3.3 summarizes the product family after the establishment of the product platform.

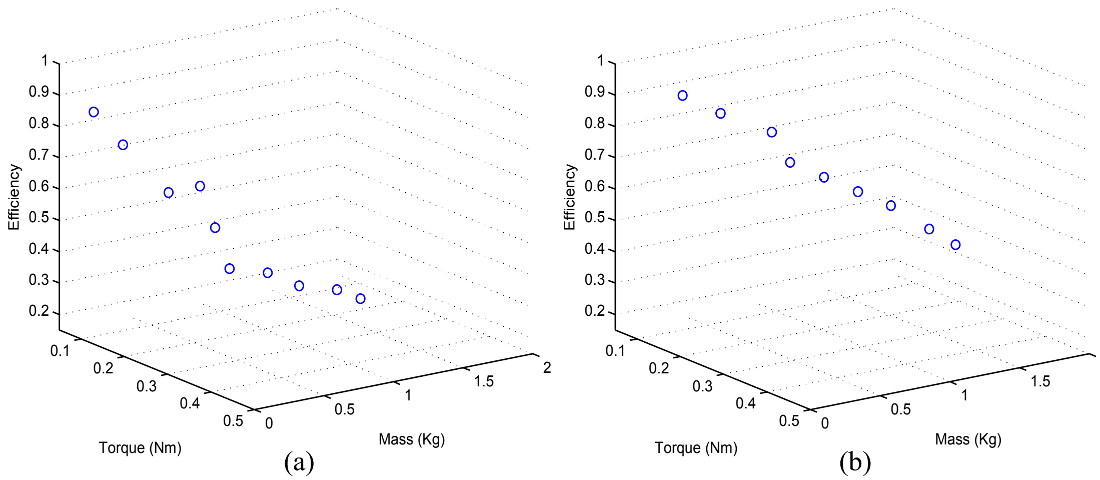


Figure 3.5: (a) Product family identified using traditional Smart Pareto filter (b) Product family identified using concurrent Smart Pareto filtering

Table 3.3: Product family using concurrent Smart Pareto filtering in variable and objective spaces after the establishment of the product platform

N_a	N_f	t	I	L_s	A_a	A_f	r	$Torque$	$Mass$	η
678	78	7.45	3.08	15.00	0.30	0.30	26.92	0.07	0.49	0.85
799	78	7.45	3.23	19.53	0.30	0.30	26.92	0.11	0.62	0.81
820	78	7.45	3.38	26.64	0.30	0.30	26.92	0.17	0.79	0.77
1095	78	7.45	3.67	21.85	0.30	0.30	26.92	0.22	0.77	0.71
1055	78	7.45	3.77	26.33	0.30	0.30	26.92	0.28	0.86	0.69
1036	78	7.45	3.88	29.93	0.30	0.30	26.92	0.33	0.94	0.67
1068	78	7.45	4.05	30.99	0.30	0.30	26.92	0.38	0.98	0.64
1108	78	7.45	4.28	31.69	0.30	0.30	26.92	0.45	1.00	0.61
1020	78	7.45	4.32	37.35	0.30	0.30	26.92	0.50	1.11	0.60
16.2%	0.00%	0.00%	11.9%	25.9%	0.00%	0.00%	0.00%			

The normalized change in family fitness per platform variable that resulted from the establishment of the platform was next calculated. First, the family fitness of the individually optimized product set, show in Table 3.2, was calculated. Similarly, the family fitness of the product family shown in Table 3.3 was also calculated. These family fitness values were 1.405 and 1.466 respectively. Also shown in Table 3.3, five platform variables were identified. The resulting percent decrease in objective performance per platform variable for the family identified using concurrent filtering was -0.87 percent, as calculated using Equation 3.8.

The effectiveness of the platform established for the benchmark family presented by Simpson et al. [12] was also calculated. The family fitness values of Simpson's individually optimized product set and product family are 1.526 and 1.615 respectively. The number of platform variables is six. The resulting percent decrease in objective performance per platform variable for the benchmark family is -0.97 percent, as calculated using Equation 3.8.

Comparing the results of concurrent Smart Pareto filtering to those of the benchmark family, two notable differences are seen. First, the percent decrease in objective performance that resulted from the implementation of the product platform was approximately 11 percent less when using concurrent filtering. This indicates that the simultaneous selection of product family members and platform features allowed for the establishment of a more efficient product platform. Second, the family fitness of the product family identified using concurrent filtering was 9.2 percent lower than the family fitness of the benchmark family. Because this problem was a minimization problem, the decrease in family fitness indicates an average improvement in the objective performance of the family identified using concurrent filtering as compared to the benchmark family. Importantly, this example illustrates the improved ability of concurrent Smart Pareto filtering to achieve high levels of commonality though minimal performance tradeoffs.

3.7 Concluding Remarks

Product families are frequently used to provide consumers with a variety of appealing products and to help maintain reasonably low production costs for manufacturers.

Three common objectives in the design of product families are used to balance the interests of both consumers and manufacturers. These objectives are to maximize product (i) performance, (ii) distinctiveness, and (iii) commonality. Concurrent Smart Pareto filtering is used to select product family members from among a candidate set consisting of many Pareto solutions. This is accomplished by considering both objective and variable spaces during the selection process. The result is a collection of many functionally optimal and distinct products that also possess limited variation in design variable space. Finally, a standard deviation to mean ratio is calculated and used in the establishment of a product platform.

Concurrent Smart Pareto filtering in variable and objective spaces is a valuable tool to help balance product performance, distinctiveness, and commonality, when designing product families with multiple functional objectives. For the universal electric motor example included in Section 3.6, the average standard deviation to mean ratio for all design variables decreased by over 35 percent, when using a concurrent Smart Pareto filter as compared to traditional Smart filters. Concurrent filtering also identified five platform variables while traditional filtering failed to identify a single platform variable.

Chapter 4

Interactive Design of Combined Scale-based and Module-based Product Family Platforms

4.1 Abstract

Product families are frequently used to provide consumers with a variety of appealing products and to help maintain reasonably low production costs for manufacturers. Three common objectives in the design of product families are used to balance the interests of both consumers and manufacturers. These objectives are to maximize (i) product performance, (ii) product distinctiveness as perceived by the consumer, and (iii) product commonality as seen by the manufacturer. To accomplish these objectives, product families are frequently designed around one of two common platform types: a scale-based product platform or a module-based product platform. In this paper, an interactive design process is presented for the design of product families built around a combined scale-based *and* module-based product platform. In the first step of this interactive process, the designer uses physical decomposition techniques to identify the major components that comprise the finished products. Second, the designer identifies and summarizes, in matrix form, the relationships that exist between each design variable and each product component. Third, multiobjective optimization is used to identify a set of many designs that are considered candidate product family members. Fourth, Smart Pareto filtering is used to select product family members from among the candidate set previously identified. A scale-based product platform is simultaneously established using concurrent Smart Pareto filtering and the component/variable relationships identified in the second step of this process. Finally, the component/variable relationships are further used to determine which product components are best suited to become modules and to determine the subset of product family mem-

bers that will use each module. In conjunction with the establishment of each module, a designer specified sequence of optimization routines is used to ensure that the three objectives of product family design, as noted above, are satisfied. As a result, a combined scale-based and module-based product platform is established. A well-known universal electric motor example problem is used to demonstrate the process.

4.2 Nomenclature

x	Vector of design variables
n_x	Number of design variables
x_c	Vector of design variables that define component interfaces
μ	Vector of design metrics (objectives)
n_μ	Number of design metrics (objectives)
g	Vector of inequality constraints
h	Vector of equality constraints
n_φ	Number of candidate product family members (solutions on discrete Pareto frontier)
$\Delta t, \Delta r$	Filter parameters defining a region of practically insignificant objective-tradeoff
Δp^i	Summation of the squared differences between each variable in product p^i and the mean of the same variable across all candidate family members
s^i	Summation of the standard deviation to mean ratios for all variables in product p^i
$\bar{x}_j^j$	Mean of design variable x_j for the product set including Smart Pareto designs and product p^i
n_ξ	Number of products in product family
ξ	Matrix of product family members prior to the establishment of the product platform
ψ	Matrix of product family members after the establishment of the product platform
J_{p^i}	Aggregate objective function for product p^i
F	Average fitness of product family
n_c	Number of individual components required to manufacture all product family members
n_i	Number of individual components required to manufacture all individually optimized product family members

n_p	Number of platform variables
C	Measure of product family commonality, ratio of n_c to n_i
N_a	Number of wire turns on armature
N_f	Number of wire turns on field, per pole
t	Thickness of stator
I	Electric current
L_s	Stack length
A_a	Armature wire cross-sectional area
A_f	Field wire cross-sectional area
r	Outer radius of stator
η	Efficiency
M	Motor mass
T	Motor torque
H	Magnetizing intensity
P_o	Gross mechanical power output
P_i	Electrical input power
M_s	Stator mass
M_a	Armature mass
M_w	Windings mass
K	Motor constant
ϕ	Magnetic flux

4.3 Introduction and Literature Survey

In today's world marketplace, consumers increasingly demand variety, customization, and personalization in the goods they purchase. For manufacturers, economically satisfying these demands is a notable challenge. In many instances, however, product families provide consumers a variety of appealing products and help manufacturers maintain reasonably low production costs through increases in economies of scale and scope [3]. Three goals, common in the design of product families, are considered when balancing the interests of both consumers and manufacturers. These goals are to maximize (i) product performance, (ii) product distinctiveness as perceived by the consumer, and (iii) product commonality as seen by the manufacturer [4, 5].

In the literature, a number of design and optimization approaches exist that address the three goals of product family design [6]. Many of these approaches identify a set of design features that are shared by product family members. In the literature, this set of shared features is known as a product family platform [34]. Traditionally, product families have been built around one of two common platform types: a scale-based product platform or a module-based product platform [35]. A scale-based platform is the foundation of related products that have differing function through the scaling of non-platform design features. A module-based platform is the foundation for a collection of related products that have differing functions through the addition or subtraction of modules. Although existing approaches address scale-based and module-based product platform design independently, the two types of product platforms are not mutually exclusive. In contrast, by carefully selecting a limited set of scale-based design features along with a complimentary set of module-based design features, a product family built around a combined scale-based and module-based product platform may require fewer total components in the manufacture of all product family members than would be required using a scale-based or module-based platform alone.

In this paper, an interactive design process is presented for the design of product families built around a combined scale-based *and* module-based product platform. In the first step of this interactive process, the designer uses physical decomposition techniques to identify the major components that comprise the finished product. Second, the designer identifies and summarizes, in matrix form, the relationships that exist between each design variable and each product component. These relationships are later used to identify the design features best suited to become scale-based or module-based platform features.

Third, multiobjective optimization is used to identify a set of many designs that are considered candidate product family members. Fourth, Smart Pareto filtering [28] is used to select product family members from among the candidate set previously identified [23]. Concurrent with the selection of the product family members, Smart Pareto filtering is also used, along with the component/variable relationships identified in the second step of this process, to identify scale-based product platform features [24]. The use of Smart Pareto filtering for the selection of product family members and the identification of scale-based platform features is of particular interest when sufficient market data is unavailable or is not easily obtained or distilled by the company developing the product family. In these cases traditional segmentation methods become difficult to implement. For example, entrepreneurs and small to mid-sized companies may not have a dedicated marketing department or the financial resources needed to purchase market research from an outside firm. When planning to launch a series of products built on a common platform, these small companies may struggle to gather and process the market data needed to construct the segmentation grid that is traditionally used to identify product family members. Smart Pareto filtering can be used in these cases for the selection of product family members as an alternative to traditional heuristic tools such as market segmentation [23].

As a final step in the interactive design process, the component/variable relationships are further used to help determine which product components are best suited to become modules and to determine the subset of product family members that will use each module. In conjunction with the establishment of each module, a designer specified sequence of optimization routines is used to ensure that the three objectives of product family design, as noted previously, are satisfied. As a result, a combined scale-based and module-based product platform is established.

The remainder of this paper is presented as follows. A preliminary review of a generic multiobjective optimization problem is included in Section 4.4. In Section 4.5, a method for establishing combined scale-based and module-based platforms is presented in its theoretical form. A universal electric motor example illustrating the interactive approach is included in Section 4.6, with our concluding remarks given in Section 4.7.

4.4 Technical Preliminary: Multiobjective Optimization

The use of the any Pareto filtering method including concurrent Smart Pareto filtering, as used in the establishment of a combined scale-based and module-based product platform, requires that a meaningful multiobjective optimization problem be formulated and solved for a Pareto frontier. A generic, mathematical description of a multiobjective optimization problem is included in Problem 1 for completeness.

Problem 1 (P1): Generic Multiobjective Optimization

$$\min_{x} \mu(x) \tag{4.1}$$

subject to:

$$g(x) \leq 0 \tag{4.2}$$

$$h(x) = 0 \tag{4.3}$$

$$x_l \leq x \leq x_u \tag{4.4}$$

When solved, multiobjective optimization problems, such as *P1*, result in many Pareto solutions that together comprise a discrete representation of the Pareto frontier. Figure 4.1 shows a hypothetical Pareto frontier for a bi-objective minimization problem. The shaded region in this figure represents feasible space, wherein all solutions satisfy every constraint. However, because both objectives in this problem are minimized, the solutions found on the lower left hand boundary of the feasible space, the Pareto solutions, are of particular interest. Importantly, each Pareto solution is non-dominated, meaning that, for each solution, no other solution exists that possesses equal or improved performance in every objective. Also, each solution is mathematically optimal, meaning that, for a single solution, any improvement in an objective can only occur at the expense of another objective [32]. Because of these important characteristics of Pareto solutions, the discrete representation of the Pareto frontier captures the tradeoffs that exist between conflicting design objectives and consists of many functionally different design alternatives.

4.5 Combined Scale-based and Module-based Platforms using Concurrent Smart Pareto Filtering

Because it consists of many optimal yet functionally different design alternatives, a Pareto frontier can be considered a set of candidate product family members. Further, a Pareto frontier can, through the use of Smart Pareto filtering, be leveraged to directly

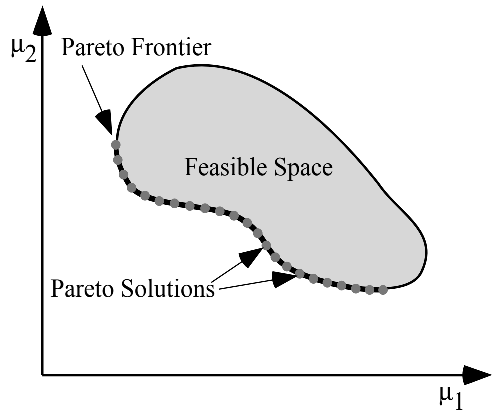


Figure 4.1: Discrete representation of Pareto frontier

satisfy two of the goals of product family design: maximum product performance and product distinctiveness. The third goal of product family design, maximum commonality, is satisfied, under the proposed method, by identifying a complimentary set of scale-based and module-based design features.

The establishment of a combined scale-based and module-based product platform is an interactive process. Each of the primary steps of this interactive process are detailed in the following sections: In Section 4.5.1, physical decomposition is used to identify each major product component. Additionally, a matrix is constructed summarizing the relationships that exist between each product component and each design variable. In Section 4.5.2, concurrent Smart Pareto filtering is used to simultaneously select product family members and identify the scale-based product platform features. Modules are then identified and the combined scale-based and module-based product platform is fully defined in Section 4.5.3. A method of comparing scale-based, module-based, and combined scale-based and module-based product platforms is presented Section 4.5.4.

4.5.1 Identifying Relationships between Design Variables and Product Components

The first step in the establishment of a combined scale-based and module-based product platform requires that the designer identify all components that together comprise

Table 4.1: Relationships between components and design variables for hypothetical product family

	x_1	x_2	x_3	x_4	x_5
$component_1$	1	0	1	1	0
$component_2$	0	0	1	1	1
$component_3$	0	1	1	0	0

an individual product family member. To accomplish this, well-known physical decomposition techniques are used. After having identified all product components, the designer then determines which design variables affect multiple components and which design variables affect a single component.

The relationships that exist between design variables and components are then summarized in matrix form where each row of the matrix represents a product component and each column represents a design variable. A value of 0 in any matrix location indicates that no relationship exists between the corresponding component and the corresponding variable; a value of 1 indicates that a relationship does exist. Table 4.1 details the relationships that exist between three product components and five design variables, for a hypothetical product family member.

Table 4.1 illustrates a number of important component/variable relationships. First, the 1 shown in the matrix location corresponding to design variable x_1 and $component_1$ indicates that $component_1$ is dependent on design variable x_1. The 0's in the matrix locations corresponding to design variable x_1 and both $component_2$ and $component_3$ indicate that neither $component_2$ nor $component_3$ is dependent on design variable x_1. Notably, because only $component_1$ depends on design variable x_1, it can be said that design variable x_1 does not define an interface between product components. Similar relationships exist between design variable x_2 and $component_3$ and design variable x_5 and $component_2$. The second important relationship illustrated in Table 4.1 is shown by the 1's in the matrix locations corresponding to design variable x_4 and both $component_1$ and $component_2$. These relationships indicate that both $component_1$ and $component_2$ depend on design variable x_4, or in other words, an interface between $component_1$ and $component_2$ is defined by design variable x_4. The third important observation that is taken from Table 4.1 is that an interface between all three components is defined by design variable x_3. This is evident from the 1's in the matrix locations corresponding to design variable x_3 and each of the product components.

4.5.2 Selecting Product Family Members and Establishing a Scale-based Product Platform

Using the Pareto frontier identified through the solution of a multiobjective optimization statement similar the one presented in Section 4.4, concurrent Smart Pareto filtering [24] is used to select the set of designs that become product family members. Simultaneously, concurrent Smart Pareto filtering is used to identify the scale-based design features that will become part of the combined scale-based and module-based product platform.

Each of the primary steps of concurrent Smart Pareto filtering are detailed in the following sections: In Section 4.5.2.1 a method for selecting a starting design is presented. In Section 4.5.2.2, the process of removing designs that are insignificantly different from other product family members is described. Section 4.5.2.3 then details the selection of subsequent designs according to the variation that would be introduced through the addition of a specific design.

4.5.2.1 Selecting a Starting Design (Single Pareto Point)

The first step of any Smart Pareto filtering process is the selection of a starting design. Traditional Smart filters typically select one of the anchor points of the Pareto frontier, a Pareto solution possessing the best possible value in a single objective, as their starting design. While the Smart set subsequently identified by the traditional Smart filter varies slightly with the selected starting design, when filtering in objective space only, the differences in the Smart sets that result from using the different anchor points as the starting design are generally small. In most cases, traditional Smart filters will produce acceptable results regardless of which anchor point is used as the starting design.

In contrast, when using concurrent Smart Pareto filtering, the selection of an anchor point as the starting design will generally yield suboptimal results. This is the case because when using concurrent filtering, the starting design directly impacts decisions made in objective *and* variable spaces. Ultimately, the poor selection of a starting design complicates the establishment of the product family platform. Therefore, when selecting a starting design for use with concurrent filtering, it is desired that the starting design be similar, at least in variable space, to the many candidate product family members. Using a design where each variable deviates minimally from the central tendency of the same variable, across all Pareto solutions, improved filtering results are obtained. Specifically, the summation of the squared differences between each variable and that variable's respective mean is used to rank each candidate starting design. The design with the smallest sum is selected as the starting design and becomes the first member of the product family. Equation 4.5 details this calculation.

$$\Delta p^i = \sum_{j=1}^{n_x} (x_j^i - \bar{x}_j)^2 \qquad (4.5)$$

The filtering process then continues, as described in Section 4.5.2.2, by removing any designs that are insignificantly different, in terms of objective performance, from the starting design presently identified. For greater detail concerning the selection of a starting design for use with concurrent Smart Pareto filtering, refer to the authors' previous publication [24].

4.5.2.2 Removing Designs of Insignificant Objective-Tradeoff

Having selected a starting design, the second step of the filtering process is to remove any designs that are insignificantly different, in terms of objective performance, from the starting design. To accomplish this, a Smart Pareto filter, defining a *Region of Insignificant Tradeoff*, is constructed and centered on the starting design, labeled $\hat{\mu}$, as shown in Figure 4.2. Pairwise comparisons are then made between the starting design and all other candidate product family members. When these comparisons identify a design that

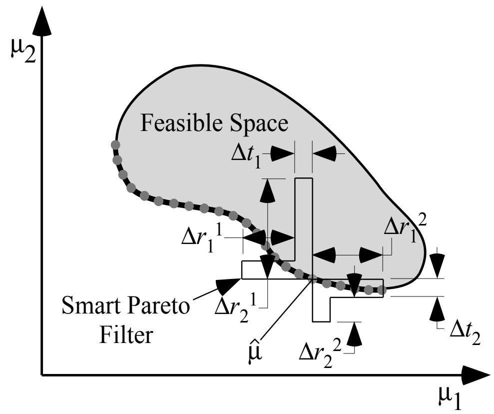


Figure 4.2: 2-D Smart Pareto filter

is located within the region of insignificant tradeoff, the design is removed from the set of candidate family members.

The geometric shape of the region of insignificant tradeoff is defined by a set of designer specified parameters, $\Delta t = \Delta t_1, \Delta t_2, ..., \Delta t_{n_\mu}$ and $\Delta r = \Delta r_1, \Delta r_2, ..., \Delta r_{n_\mu}$, where $\Delta r_i = \Delta r_i^1, \Delta r_i^2, ..., \Delta r_i^{i-1}, \Delta r_i^{i+1}, ..., \Delta r_1^{n_\mu}$, for $i = 1...n_\mu$. Figure 4.2 shows each of these parameters for the bi-objective case. The two parameters, Δt and Δr can be understood by the following; when comparing two Pareto points (or designs), any difference in μ_i that is less than Δt_i is considered to be insignificant. As such, one of the two designs being compared should be removed since it is insignificantly different – at least in one objective. When, however, the difference in μ_i, between two points, is within Δt_i, but the change in another objective (μ_j) is significantly large (greater than Δr_j), then the design should not be removed. In other words, any change in μ_j that is larger than Δr_j is deemed significant – regardless of how small the change in another objective is. In this way, the parameters Δt and Δr are physically meaningful and can be easily specified by the designer.

Following the removal of all candidate product family members that are insignificantly different from the starting design, the filtering process then continues, as described in Section 4.5.2.3, with the selection of subsequent product family members. For greater detail concerning the removal of designs of insignificant objective tradeoff, refer to the authors' previous publication [24].

4.5.2.3 Selecting Subsequent Smart Set Designs

When using concurrent Smart Pareto filtering for the establishment of a combined scale-based and module-based product platform, each subsequent design is selected according to the impact that its addition to the product family has on the summation of standard deviation to mean ratios, taken across all product family members, for a subset of design variables. Importantly, the subset of design variables considered during the selection of subsequent product family members is limited to the set of design variables that define an interface between product components. The component/variable matrix defined in Section 4.5.1 is used to identify the design variables included in this set. The design resulting in the smallest sum is selected as the next member of the product family. Equations 4.6 and 4.7 summarize the calculation used to rank each design. For greater detail concerning the selection of subsequent members of the product family, refer to the authors' previous publication [24].

$$s^i = \sum_{j=1}^{n_x} \frac{[(\sum_{k=1}^{n_\xi}(\xi_j^k - \bar{x}_j^i)^2 + (x_j^i - \bar{x}_j^i)^2)/n_\xi]^{1/2}}{\bar{x}_j^i} \tag{4.6}$$

Where:

$$x_j \in x_c \tag{4.7}$$

Finally, because the starting design as well as the subsequent designs that become members of the product family are selected so as to minimize the summation of standard deviation to mean ratios for a subset of design variables, concurrent Smart Pareto filtering minimizes variation among design variables that define component interfaces. Considering each of these design variable individually, any design variable with a standard deviation to mean ratio of less than 10 percent is considered well suited as a platform variable [37]. A single optimization routine is then used to eliminate all variation from platform variables and to adjust non-platform variable values so as to minimize changes in objective performance. The result is the identification of the scale-based design features that will be included in the combined scale-based and module-based product platform.

Table 4.2: Design variable values of hypothetical product family members after scale-based platform implementation

	x_1	x_2	x_3	x_4	x_5
$product_1$	5.51	4.63	2.25	1.52	6.41
$product_2$	5.42	6.77	2.25	1.52	7.13
$product_3$	3.44	4.52	2.25	1.52	9.54
$product_4$	3.37	6.84	2.25	1.52	8.78
$product_5$	3.41	5.82	2.25	1.52	7.87
$product_6$	5.46	5.70	2.25	1.52	8.36

4.5.3 Establishing a Module-based Platform

Following the identification of scale-based design features, including the identification of platform variables, modules are identified by considering non-platform design variables over discrete product ranges. The discrete ranges are selected so that the standard deviation to mean ratio, for each variable in a subset of design variables, is less than 10 percent. To illustrate the process, a hypothetical product family with five design variables and six members is considered. The design variable values for this family, after the establishment of a scale-based platform, are included in Table 4.2.

The products in this family are made up of three major physical components each of which could become a module. Table 4.1, as shown in Section 4.5.1, details the relationship between each component and each design variable. Each value of 1 shown in Table 4.1 indicates that the corresponding component is dependent on the corresponding variable, while a 0 indicates that no dependence exists between that module and variable. Following the approach outlined in Section 4.5.2.3, design variables x_3 and x_4 have become platform variables, common to all products in the family. The establishment of this scale-based platform improves the possibility of identifying a complimentary set of modules following the process described below.

By considering discrete groups of products, in conjunction with the component/variable matrix shown in Table 4.1, modules are identified. First, from Table 4.1, it is noted that $component_1$ is dependant on design variables x_1, x_3, and x_4, as indicated by the 1 shown

Table 4.3: Design variable values of hypothetical product family members after combined scale-based and module-based platform implementation

	x_1	x_2	x_3	x_4	x_5
$product_1$	5.47	4.58	2.25	1.52	6.41
$product_2$	5.47	6.81	2.25	1.52	7.13
$product_3$	3.40	4.58	2.25	1.52	9.54
$product_4$	3.40	6.81	2.25	1.52	8.78
$product_5$	3.40	5.77	2.25	1.52	7.87
$product_6$	5.47	5.77	2.25	1.52	8.36

in each of the corresponding columns. Further, for products 1, 2, and 6 it is noted that the standard deviation to mean ratios for variables x_1, x_3, and x_4 are each less than 10 percent, as shown in Table 4.2. Likewise, the standard deviation to mean ratios for variables x_1, x_3, and x_4 for products 3, 4, and 5 are less than 10 percent. From this it is concluded that, for all members of the product family, one of two modules will satisfy each product's design need for $component_1$. Similarly, from Table 4.1, it is noted that $component_3$ is dependent on variables x_2 and x_3. For products 1 and 3, 2 and 4, and 5 and 6, taken each in turn, the standard deviation to mean ratio for variables x_2 and x_3 are each less than 10 percent. This indicates that one of three modules will adequately satisfy each product's design need for $component_3$. Finally, because the standard deviation to mean ratio for design variable x_5 is greater than 10 percent for all product family member combinations, $component_2$ does not become a module; design variable x_5 thus remains a scalable variable. The establishment of these modules results in the design variable values shown in Table 4.3. Figure 4.3 shows the hypothetical product family that results from the process.

4.5.4 Measuring Family Performance and Commonality

After defining a product platform, we seek to classify the performance and commonality changes resulting from the establishment of the product family. To do this, the Smart Pareto set of individually optimized products and the platform-based product family are compared. The change in both objective performance and product commonality can

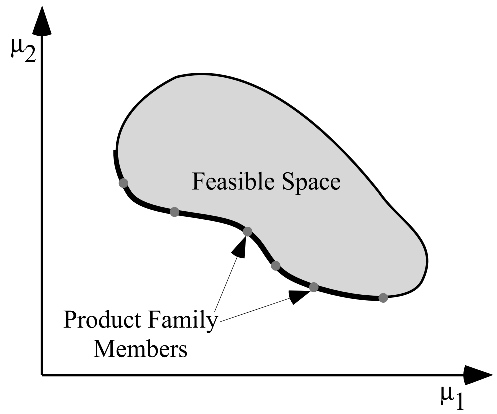


Figure 4.3: Product family resulting from concurrent Smart Pareto filtering

be evaluated numerically by calculating a *family fitness* value. Equation 4.8 details the calculation. Note that this measure of family fitness considers the effect that establishing a platform has on product performance and product commonality but does not consider diversity in its calculation. Such considerations are the focus of future work.

$$F = \sum_{i=1}^{n_\xi} (J_{p^i})/n_\xi + C \tag{4.8}$$

$$C = n_c/n_i + (n_x - n_p)/n_x \tag{4.9}$$

In the case of a minimization problem, a smaller family fitness value corresponds to a higher average performance for the product family. The first term in the family fitness calculation exclusively considers the objective performance of the product family. The second term of the family fitness equation calculates product commonality. This calculation is detailed in Equation 4.9. In the case of the hypothetical product family presented in Section 4.5.3, two instances of $component_1$, six instances of $component_2$, and three instances of $component_3$ are required for the manufacture of the six product family members, or 11 distinct components in total. This is compared to the 18 distinct components required for the manufacture of the individually optimized products. Further, two of the five design variables were determined to be platform variables. The resulting commonality calculation results in a product commonality of 1.21.

In the section that follows, Equations 4.8 and 4.9 are used to compare a product family built on a combined scale-based and module-based platform to a benchmark family of universal electric motors [12].

4.6 Universal Electric Motor Example Problem

The electric motor example problem included in this section is a tri-objective problem based on the well-known bi-objective problem [33]. In this example, given a set of filter parameters, concurrent Smart Pareto filtering in variable and objective spaces is used to select product family members and to identify the scale-based design features that are included in the combined scale-based and module-based platform. Non-platform variables are then considered over discrete product ranges and a complementary set of modules is identified.

The process begins with the physical decomposition of the product and the identification of the relationships that exist between each product component and each design variable. The universal electric motor consists of two major components: an armature and a field. Eight design variables define the characteristics of each product. Table 4.4 details the component/variable relationships that will be used in subsequent steps of this example problem.

Table 4.4: Electric motor component dependencies by design variable

	N_a	N_f	t	I	L_s	A_a	A_f	r
Armature	1	0	0	0	1	1	0	1
Field	0	1	1	0	1	0	1	1

After identifying the component/variable relationships, the interactive process continued with the solution of the multiobjective optimization problem statement detailed in Problem 2. The result was a set of many functionally different electric motors that were considered candidate product family members.

Problem 2 (P2): Multiobjective Optimization of a Family of Electric Motors

$$\min_{x}[-T(x) \quad M(x) \quad -\eta(x)]^T \tag{4.10}$$

subject to:

$$\begin{aligned}
t-r &< 0 && (4.11)\\
H-5000 \text{ Ampere/m} &< 0 && (4.12)\\
0.15-\eta &\leq 0 && (4.13)\\
M-2.0 \text{ Kg} &\leq 0 && (4.14)\\
0.05 \text{ Nm}-T &\leq 0 && (4.15)\\
T-0.5 \text{ Nm} &\leq 0 && (4.16)\\
P_o-300 \text{ W} &= 0 && (4.17)
\end{aligned}$$

where:

$$\begin{aligned}
x &= [N_a, N_f, t, I, L_s, A_a, A_f, r] && (4.18)\\
\eta &= P_o/P_i && (4.19)\\
M &= M_s+M_a+M_w && (4.20)\\
T &= K\phi I && (4.21)
\end{aligned}$$

Additional equations used in the derivation of the objective functions are detailed in the problem testbed [33].

P2 was solved using a Normal Constraint [32] optimization approach. Figure 4.4(a) shows a discrete representation of the resulting Pareto frontier. Using the set of filter parameters detailed below along with concurrent Smart Pareto filtering, the discrete representation of the Pareto frontier was reduced to the set of product family members shown in Figure 4.4(b).

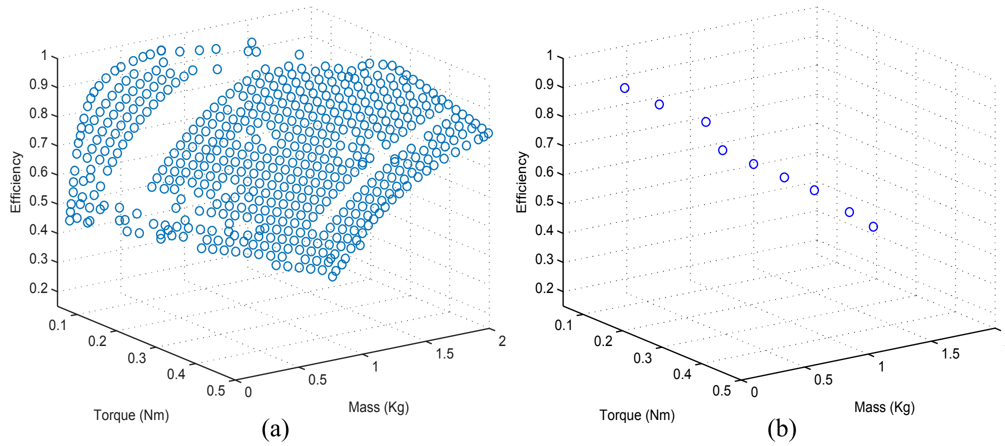


Figure 4.4: (a) Discrete representation of Pareto frontier (b) Product family identified using concurrent Smart Pareto filtering

(1)$\Delta\eta \leq 0.05$ is insignificant, an increase $\Delta\eta > 0.12$ is notable and no feasible decrease in η is considered noteworthy.

(2)$\Delta M \leq 0.25$ Kg is insignificant, a decrease $\Delta M > 0.50$ Kg is notable, and no feasible increase in mass is considered noteworthy.

(3) $\Delta T \leq 0.03$ Nm is insignificant and an increase or decrease $\Delta T > 0.04$ Nm is notable.

Table 4.5 shows the design variable values for each of the nine products that were identified to be part of the product family. Using equation 4.8 family fitness was measured prior to the implementation of the platform. The set of individually optimized product family members had fitness of 3.4640.

Concurrent Smart Pareto filtering was also used to simultaneously identify the scale-based platform features. As described in Section 4.5.2.3, preference was given to interface variables during the establishment of the scale-based platform. The percentages at the bottom of Table 4.5 are the standard deviation to mean ratios for each of the design variables. Importantly, the design variables L_s and r each have a standard deviation to mean ratio of less than 10 percent, indicating that each of these interface variables is well suited to become a platform variable. Additionally, the design variable t was also identified as well

Table 4.5: Design variable values of product family members prior to the implementation of a product platform

N_a	N_f	t	I	L_s	A_a	A_f	r
681	96	7.15	2.84	22.16	0.56	0.56	21.88
697	107	7.22	2.96	25.17	0.47	0.47	25.33
1000	84	6.49	3.48	24.48	0.29	0.29	23.41
985	84	6.79	3.71	25.39	0.28	0.24	24.82
997	71	6.57	4.40	25.98	0.24	0.19	24.74
1091	64	6.67	4.90	24.91	0.24	0.24	25.10
1086	57	6.82	5.54	24.81	0.23	0.21	25.17
1081	89	8.57	4.20	26.91	0.29	0.29	29.79
1155	54	6.89	5.97	26.13	0.25	0.25	25.62
17.6%	23.1%	9.0%	25.9%	5.3%	37.1%	41.3%	8.4%

suited to become a platform variable. The other five design variables remained scalable variables.

Table 4.6 shows the design variable values of each of the nine product family members after the implementation of the scale-based platform. The family fitness for the scale-based product family was 3.1624. The decrease of 0.3016 in family fitness, as compared to the individually optimized set of product family members, indicates that the increases in product commonality were relatively large when compared to the performance decreases that resulted from the establishment of the scale-based product family.

To further increase product commonality, subsets of non-platform or scalable variables were considered over discrete ranges of product family members. First, considering the armature, products 1 and 2, 3 and 4, 5 and 6, 7, 8, and 9 were included in individual product subsets. For the field, products 1, 2-4, and 5-9 were likewise grouped in product subsets. At this point, an interactive series of designer specified optimization routines were used to eliminate all variation from within each product subset for each component. Notably, during the optimization routines, the designer recognized that for the field component of product 5, the design variable N_f tended toward a much lower value. As a result, as part of the interactive design process, the designer reconfigured the product subsets so that products 1 and 5, 2-4, and 6-9 were then grouped. A designer specified series of opti-

Table 4.6: Design variable values of product family members after the implementation of a scale-based product platform

N_a	N_f	t	I	L_s	A_a	A_f	r
500	60	6.93	3.23	26.91	0.24	0.24	28.98
500	94	6.93	3.36	26.91	0.23	0.23	28.98
560	99	6.93	3.69	26.91	0.21	0.21	28.98
646	94	6.93	3.90	26.91	0.21	0.21	28.98
699	83	6.93	4.42	26.91	0.20	0.20	28.98
784	77	6.93	4.75	26.91	0.21	0.21	28.98
879	77	6.93	4.75	26.91	0.23	0.23	28.98
971	77	6.93	4.75	26.91	0.25	0.25	28.98
1100	77	6.93	4.75	26.91	0.28	0.28	28.98
28.8%	14.6%	0.0%	15.3%	0.0%	11.2%	11.2%	0.0%

mization routines followed until all variation was eliminated from each product subset for each component, resulting in six armature modules and three field modules. The design variable values for the product family with a combined scale-based and module-based platform are shown in Table 4.7. The product family with the combined platform had fitness of 2.7780. This notably lower family fitness value indicates that the product commonality and performance changes resulting from the establishment of the combined scale-based and module-based product platform have improved the overall characteristics of the product family.

In order to compare the results of the combined scale-based and module-based product platform to a benchmark product family [12], the family fitness of the benchmark family was calculated, resulting in a value of 2.8652. Importantly, the benchmark family did not identify a single module, but did identify six of eight design variables as platform variables. Notably, the family fitness of the product family built on the combined scale-based and module-based platform was 3.1 percent lower that the family fitness of the benchmark family. Because the universal electric motor problem detailed in this section is a minimization problem, the slightly lower family fitness value of the combined platform corresponds to an improvement over the benchmark family.

Table 4.7: Design variable values of product family members after the implementation of a combined scale-based and module-based product platform

N_a	N_f	t	I	L_s	A_a	A_f	r
499	59	6.93	3.26	26.91	0.24	0.19	28.98
499	65	6.93	4.02	26.91	0.24	0.07	28.98
455	65	6.93	5.04	26.91	0.18	0.07	28.98
455	65	6.93	5.56	26.91	0.18	0.07	28.98
815	59	6.93	4.83	26.91	0.21	0.19	28.98
815	75	6.93	4.71	26.91	0.21	0.25	28.98
880	75	6.93	4.81	26.91	0.23	0.25	28.98
970	75	6.93	4.81	26.91	0.25	0.25	28.98
1100	75	6.93	4.81	26.91	0.28	0.25	28.98

4.7 Concluding Remarks

In many cases, a combined scale-based and module-based product platform may reduce the number of components required for the manufacture of a family of related products. In this paper, an interactive method for establishing a combined scale-based and module-based platform is presented. The method begins with the use of physical decomposition in the identification of all major product components. Next, the designer identifies the relationships that exist between each product component and each design variable. Third, the process continues with the use of concurrent Smart Pareto filtering to identify product family members and to establish a scale-based platform for those products. Fourth, non-platform variables are considered over discrete product ranges and modules are established. Finally, measures for both objective performance and product commonality are used to assess the effectiveness of the product platform. The method is demonstrated by designing a family of universal electric motors.

Chapter 5

Conclusion

5.1 Conclusions

In this thesis, three methods are introduced that use multiobjective optimization and Smart Pareto filtering to satisfy the three objectives of product family design, namely to maximize (i) product performance, (ii) product distinctiveness as perceived by the consumer, and (iii) product commonality as seen by the manufacturer. The methods are progressive in nature, building on each other in concept and complexity. Each of the three methods follows a common design approach. This common approach is summarized in the following three step process:

> (1) Objective performance of each product family member is maximized through the use of multiobjective optimization.
> (2) Product distinctiveness is ensured by using Smart Pareto filters for the selection of product family members.
> (3) One of three distinct methods of establishing product platforms is used to maximize product commonality.

In the first of the three methods, Smart Pareto filtering is introduced as a means of determining the number of members to include in a product family, identifying the members themselves, and defining the product platform. The use of Smart filters is particularly valuable when traditional market segmentation data is unavailable or is not easily gathered or distilled by the company designing the product family. According to the methodology, (1) a discrete representation of a Pareto frontier comprised of candidate family members is identified. (2) The filter is then used to identify a Smart Pareto set of products that constitutes the product family. This Smart Pareto set includes the members needed to make a product family sufficiently diverse and represents the least number of products that provide adequate representation of product performance tradeoffs. (3) A designer-specified maximum allowable difference in each design variable, threshold of variation, is used to determine which of the design variables are best suited as platform variables and which are best suited as scalable variables.

In the second of the three methods introduced in this work, concurrent Smart Pareto filtering is used to select product family members from among a candidate set consisting of many Pareto solutions. Notably, concurrent filtering considers both objective and variable spaces during the selection process. The filter functions as follows: (1) Given a set of Pareto solutions, a starting design is identified and classified as the first member of the product family. (2) A region of insignificant tradeoff is constructed about the starting design in objective space. Any design located within the region of insignificant tradeoff is eliminated from the set of candidate product family members. (3) Each remaining candidate product family member is ranked according to the effect that its addition to the product family would have on the variation among design variables. The design resulting in the lowest calculated standard deviation to mean ratio, summed across all design variables, is selected as the next member of the product family. (4) A region of insignificant tradeoff is constructed about this design and any designs located within this space are removed from the set of candidate product family members. Steps 3 and 4 are then repeated until all points have either been included as a member of the product family or identified as being insignificantly different from at least one product family member. (5) A standard deviation to mean ratio is calculated for each design variable and any variable with a ratio of less than 10 percent is made a platform variable.

The third and last method introduced in this work presents an interactive method for establishing a combined scale-based and module-based product platform. The method begins with (1) the use of physical decomposition in the identification of all major product components. (2) Next, the designer identifies the relationships that exist between each product component and each design variable. (3) The process then continues with the use

Table 5.1: Comparison of the results achieved through each of the introduced methods and the benchmark family of universal electric motors

	Objective Performance	Commonality	Aggregate Total
Benchmark Set	1.526	2.000	3.526
Benchmark Family	1.615	1.250	2.865
Traditional Smart Pareto Filter	1.548	1.750	3.298
Concurrent Smart Pareto Filter	1.467	1.375	2.842
Combined Scale-based/Module-based	1.653	1.125	2.778

of concurrent Smart Pareto filtering to identify product family members and to establish a scale-based platform for those products. (4) Non-platform variables are next considered over discrete product ranges and modules are established.

Table 5.1 shows the objective performance and commonality measures for each of the methods introduced in this work as well as the individually optimized set of benchmark products and the benchmark product family. Notably, the method resulting in the best objective performance (the lowest value in the case of a minimization problem) was concurrent Smart Pareto filtering. The method resulting in the highest level of product commonality (again noted by the lowest value) was the combined scale-based and module-based product platform. Each of the methods introduced in this work showed improvement over the benchmark set of individually optimized products, as indicated by the corresponding aggregate totals. This indicates that the decrease in objective performance that resulted from the implementation of the respective product platform was relatively small in comparison to the improvement in product commonality. Importantly, the benchmark family showed a similar improvement in its aggregate total when compared to the benchmark set of individually optimized products. Also of note, both concurrent Smart Pareto filtering and the combined scale-based and module-based platform approach showed improvements over the benchmark product family.

As noted in Section 1.1, the objective of this thesis is the development of an approach that aids designers in determining product family size, members, and platform, including the identification of modular components and scalable parameters to be used in the design of a combined module-based and scale-based product family. The research detailed in the three publications that constitute the body of this thesis collectively satisfy this objective. Chapter 2 presents Pareto frontiers as candidate sets of product family members and introduces the use of Smart Pareto filters for the selection of family members. Chapter 3 builds upon the use of Smart Pareto filters by concurrently searching design variable space, allowing for the simultaneous selection of product family members and establishment of the product platform. Chapter 4 then applies concurrent Smart Pareto filtering in the design of product families built upon a combined scale-based and module-based platform.

At this point, a note on the use of each method is in order. First, of the three methods introduced in Chapters 2 through 4, concurrent Smart Pareto filtering and combined scale-based and module-based platform design are most useful. That being said, the threshold of variation and Smart Pareto filtering method detailed in Chapter 2 is of value in that it bridges the gap between research in multiobjective optimization techniques and product family design. Further, Chapter 2 lays a groundwork for the developments detailed in Chapters 3 and 4. Second, concurrent Smart Pareto filtering, as detailed in Chapter 3, is limited in use to product families that can successfully be built on a purely scale-based product platform. Third, concurrent Smart Pareto filtering can successfully and relatively easily be used in the design of product families with any number of objectives and/or design variables, as long as a Pareto set of optimal candidate solutions can be generated. In the literature, Pareto set generators are available that can identify many Pareto solutions (tens of thousands) from problems consisting of hundreds of design variables, objectives, and constraints. The scale of the problem has only minor impacts on the usefulness of the method. Fourth, combined scale-based and module-based platform design, as presented in Chapter 4, is, in contrast, severely limited by the scale of the design problem at hand. The difficulties limiting its use principally arise from the high number of design variable/component interactions that exist in large design problems. It is important to note that even for the relatively simple universal electric motor example problem presented in Chapter 4, which had only 16 design variable/component interactions, a significantly higher level of complexity was observed in arriving at a solution using the combined platform approach as compared to concurrent filtering. For design problems consisting of dozens of components and hundreds of design variables, the number of interactions could easily increase 10 fold or more, perhaps making the resolution of all variable/component interactions impossible. Finally, when selecting between the two methods, preference should be given to concurrent Smart Pareto filtering for the design of complex product families. Combined scale-based and module-based design would in turn be preferred for the design of product families requiring a particularly high level product commonality.

Further, for each of the methods presented within the body of this thesis a few final points should be considered. First, the successful identification of a set of product family members is somewhat limited by the Pareto frontier that is identified by the optimization algorithm. The designer should exercise care so as to select an optimization algorithm that is well suited for the problem at hand. Second, Smart Pareto filters are relatively unaffected by disjointedness or sparseness along the Pareto frontier, allowing for the selection of product family members in otherwise difficult circumstances. Finally, for concurrent Smart Pareto filters to function properly, each design must share a common set of design variables. In other words, Pareto frontiers that are composed of multiple Pareto sets cannot necessarily be used in conjunction with concurrent Smart Pareto filtering.

5.2 Future Work

In considering opportunities for future work, three major areas were identified. First, both ε-Constraint and Normal Constraint methods were used in the example problems to identify the candidate set of products that made up each Pareto frontier. Each of these optimization methods identifies the various Pareto solutions by completing, in a systematic way, many single objective optimization routines. In certain cases, an individual optimization routine can identify an area within the objective space that contains no valid Pareto solutions. The current systematic approaches used in both ε-Constraint and Normal Constraint will continue to search this space, even though no solutions will be found, increasing computational time significantly. Improvements to the systematic approaches used to identify the many Pareto solutions could be made so that any space that contains no solutions is search only once. These improvements would increase the efficiency of the three methods introduced in this work. Further, because a Smart Pareto filter is used to remove overly similar designs from among the candidate set, many of the solutions identified during the optimization routine are ultimately discarded. Additional improvements in efficiency could be achieved by further refining the systematic approach of each optimization algorithm so that candidate designs are only identified in areas where significant differences in objective performance exists. This Smart generation of designs could ultimately eliminate the need for filtering altogether.

Second, it is noted that in some cases, product families will require highly varying spaces between products in the product family along one or more objective axes. Such variability may be required when tightly grouped subsets of products can be used, for example, to discourage competitors from entering a particular market segment. The filtering methods presented in this work do not provide a means for the designer to specify filter parameters as a function of position along an objective axis and therefore does not generally result in highly varying spaces between family members, although it can happen naturally depending on the objective functions scales and the nature of the Pareto frontier. This type of flexibility in specifying filter parameters would allow the filter to identify product families with highly varying spaces between products in that family. The development of this flexibility could be considered as next phase of research in this area.

Third, while the use of Smart Pareto filters in the selection of product family members lends itself to the design of product families in the absence of market data, it is not limited to such cases. An advanced marketing tool, conjoint analysis, could be used to defined the tradeoffs that exist among design objectives. By defining these tradeoffs, conjoint analysis in essence defines the parameters of the Smart Pareto filter as well. Formalizing the link between market research, using conjoint analysis, and the selection of product family members, using Smart Pareto filters, is an important next step of this research. This link would improve the usefulness of the methods introduced in this work when designing a product family in a well defined market niche.

References

[1] Clemons, E. K., and Spitler, R., 2003. "I'm entitled! or profiting from customer preferences in a resonance economy." In *Working Paper* Draft 6.4, 18 December.

[2] Pine, B. J., 1993. *Mass Customization: The New Frontier in Business Competition*. Harvard Business Press, Boston, Ma.

[3] Tseng, M. M., and Jiao, J., 1998. "Design for mass customization by developing product family architecture." In *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference* Paper No.DETC98/DFM-5717.

[4] Fellini, R., Kokkolaras, M., and Papalambros, P. Y., 2003. "A rigorous framework for making commonality and modularity decisions in optimal design of product families." In *International Conference on Engineering Design*.

[5] de Weck, O. L., Suh, E. S., and Chang, D., 2003. "Product family and platform portfolio optimization." In *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference* Paper No.DETC03/DAC-48721.

[6] Simpson, T. W., 2003. "Product platform design and optimization: Status and promise." *Artifcial Intelligence for Engineering Design, Analysis and Manufacturing,* **18**(1), pp. 3–20.

[7] Fellini, R., Kokkolaras, M., , Papalambros, P., and Perez-Duarte, A., 2002. "Platform selection under performance loss constraints in optimal design of product families." In *ASME Design Engineering Technical Conferences - Design Automation Conference* Paper No. DETC2002/DAC-34099.

[8] K. Fujita, S. Akagi, T. Y. M. I., 1998. "Simultaneous optimization of product family sharing system structure and configuration." In *ASME Design Engineering Technical Conferences - Design for Manufacturing Conference* Paper No. DETC98/DFM-5722.

[9] Cetin, O. L., and Saitou, K., 2004. "Decomposition-based assembly synthesis for structural modularity." *ASME Journal of Mechanical Design,* **126**, pp. 234–243.

[10] D'Souza, B. S., and Simpson, T. W., 2003. "A genetic algorithm based method for product family design optimization." *Engineering Optimization,* **35**(1), pp. 1–18.

[11] Gonzalez-Zugasti, J. P., and Otto, K. N., 2000. "Modular platform-based product family design." In *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference* Paper No.DETC2000/DAC-14238.

[12] Simpson, T., Maier, J., and Mistree, F., 2001. "Product platform design: Method and application." *Research in Engineering Design,* **13**(1), pp. 2–22.

[13] Farrell, R., and Simpson, T. W., 2003. "Product platform design to improve commonality in custom products." *Journal of Intelligent Manufacturing,* **14**(6), pp. 541–556.

[14] T. S. Chang, A. C. W., 1995. "Design-in-modularity with conceptual robustness." *ASME,* **82**(1), pp. 493–500.

[15] Blackenfelt, M., 2000. "Profit maximisation while considering uncertainty by balancing commonality and variety using robust design - the redesign of a family of lift tables." In *ASME Design Engineering Technical Conferences - Design for Manufacturing* Paper No. DETC2000/DFM-14013.

[16] Fujita, K., and Yoshida, H., 2004. "Product variety optimization simultaneously designing module combination and module attributes." *Concurrent Engineering,* **12**(2), pp. 105–118.

[17] G. Hernandez, J. K. Allen, F. M., 2003. "Platform design for customizable products as a problem of access in a geometric space." *Engineering Optimization,* **35**(3), pp. 229–254.

[18] Meyer, M. H., and Lehnerd, A. P., 1997. *The Power of Product Platforms: Building Value and Cost Leadership*. Free Press, New York, NY.

[19] Dai, Z., and Scott, M. J., 2006. "Effective product family design using preference aggregation." *Journal of Mechanical Design,* **128**(4), pp. 659–667.

[20] Seepersad, C. C., Hernandez, G., and Allen, J. K., 2000. "A quantitative approach to determining product platform extent." In *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference* Paper No.DETC2000/DAC-14288.

[21] Kumar, D., Chen, W., and Simpson, T., 2006. "A market-driven approach to the design of platform-based product families." In *AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* Paper No.AIAA-2006-6919.

[22] Michalek, J., Ceryan, O., Papalambros, P., and Koren, Y., 2006. "Balancing market and manufacturing objectives in product line design." *Journal of Mechanical Design,* **128**(6), pp. 1196–1204.

[23] Yearsley, J. D., and Mattson, C. A., 2008. "Product family design using a smart pareto filter." In *46th AIAA Aerospace Sciences Meeting and Exhibit* Paper No. AIAA-2008-0909.

[24] Yearsley, J. D., and Mattson, C. A., 2008. "Product family member selection and platform variable identification: Concurrent smart pareto filtering in variable and objective spaces." In *AIAA Multidisciplinary Design Optimization Specialist Conference* Paper No. AIAA-2008-2220.

[25] Yearsley, J. D., and Mattson, C. A., 2008. "Interactive design of combined scale-based and module-based product family platforms." In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* Paper No. AIAA-2008-5819.

[26] Gonzalez-Zugasti, J. P., Otto, K. N., and Baker, J. D., 2000. "A method for architecting product platforms." *Research in Engineering Design,* **12**, pp. 61–72.

[27] Robertson, D., and Ulrich, K., 1998. "Planning product platforms." *Sloan Management Review,* **39**(4), pp. 19–31.

[28] 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.

[29] Khire, R. A., Messac, A., and Simpson, T. W., 2006. "Optimal design of product families using slection-integrated optimization methodology." In *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* Paper No. AIAA-2006-6924.

[30] Rai, R., and Allada, V., 2003. "Modular product family design: Agent-based pareto-optimization and quality loss function-based post-optimal analysis." *International Journal of Production Research,* **41**(17), pp. 4098–4075.

[31] Nelson, S. A., Parkinson, M. B., and Papalambros, P. Y., 2001. "Multiccriteria optimization in product platform design." *Journal of Mechanical Design,* **123**, pp. 199–204.

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

[33] Scott, M. J., Allada, V., and Simpson, T. W., 2006. Problem family design optimization testbed See URL http://edog.mne.psu.edu/platform/testbed/, December.

[34] Halman, J. I. M., Hofer, A. P., and v. Vuuren, W., 2003. "Platform-driven development of product families: Linking theory with practice." *Journal of Product Innovation Management,* **20**, pp. 149–162.

[35] Yang, T. G., Beiter, K. A., and Ishii, K., 2004. "Product platform development: An approach for products in the conceptual stages of design." In *2004 ASME International Mechanical Engineering Congress and RDD Expo* Paper No. IMECE2004-62171.

[36] Dai, Z., and Scott, M. J., 2005. "Meaningful tradeoffs in product family design positive and negative aspects of commonality." In *SAE Special Publication SP-1956* Paper No. 2005-01-1351.

[37] Nayak, R., Chen, W., and Simpson, T., 2002. "A variation based method for product family design." *Engineering Optimization,* **34**(1), pp. 65–81.