Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2007
September 4-7, 2007, Las Vegas, Nevada, USA

DETC2007-35611

A SINGLE-STAGE GRADIENT-BASED APPROACH FOR SOLVING THE JOINT PRODUCT FAMILY PLATFORM SELECTION AND DESIGN PROBLEM USING DECOMPOSITION

Aida Khajavirad
Research Assistant

Jeremy J. Michalek
Assistant Professor

Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

A core challenge in product family optimization is to develop a single-stage approach that can optimally select the set of variables to be shared in the platform(s) while simultaneously designing the platform(s) and variants within an algorithm that is efficient and scalable. However, solving the joint product family platform selection and design problem involves significant complexity and computational cost, so most prior methods have narrowed the scope by treating the platform as fixed or have relied on stochastic algorithms or heuristic two-stage approaches that may sacrifice optimality. In this paper, we propose a single-stage approach for optimizing the joint problem using gradient-based methods. The combinatorial platform-selection variables are relaxed to the continuous space by applying the commonality index and consistency relaxation function introduced in a companion paper. In order to improve scalability properties, we exploit the structure of the product family problem and decompose the joint product family optimization problem into a two-level optimization problem using analytical target cascading so that the system-level problem determines the optimal platform configuration while each subsystem optimizes a single product in the family. Finally, we demonstrate the approach through optimization of a family of ten bathroom scales; Results indicate encouraging success with scalability and computational expense.

KEYWORDS: Product Family, Single-Stage Approach, Platform Selection, Scalability, Decomposition.

NOMENCLATURE

CI: Commonality Index
$\mathbf{f}^i$: Objective function vector for the i^{th} product
$\mathbf{h}^i$: Vector of equality constraints for the i^{th} product
$\mathbf{g}^i$: Vector of inequality constraints for the i^{th} product
S_{ij}: Platform configuration index set
s_k,: Number of distinct platforms for producing the k^{th} component
u: Total number of distinct components in the product family
$\mathbf{x}^i$: Design variable vector for the i^{th} product
α: Relaxation factor
η^{ij}_{pq} : Binary commonality decision variables
$\tilde{\lambda}^k_r$: r^{th} eigenvalue of the k^{th} commonality matrix (including zero and nonzero terms)
λ^k_r: r^{th} nonzero eigenvalue of $\mathbf{\Gamma}_k$
Γ: Commonality objective function for the entire family
$\mathbf{\Gamma}_k$: Commonality matrix for the k^{th} component

1. INTRODUCTION

A product family can be defined as a group of related products derived from a number of shared components produced in the same platform. One main challenge in designing a successful product family is to exploit commonality for decreasing manufacturing cost without sacrificing the required distinctiveness for attracting a variety of market segments. While increasing the number of common modules among variants in the product family generally reduces cost, it also leads to some loss in the ability to achieve individual performance targets. Hence, resolving the tradeoff between commonality and the ability to achieve distinct performance targets has been the focus of many studies during the past decade.

Simpson *et al*. [1] reviews and compares forty approaches addressing the product family optimization problem. According to this classification, some methods limit scope in order to reduce complexity by assuming that design variables defining product platforms are known *a priori* and are not treated as variables in the optimization process (Allada and Jiang [2]; Blackenfelt [3], D'souza and Simpson [4], Dai and Scott [5], Farrell and Simpson [6], Fellini *et al*. [7],;Gonzales-Zugasti *et al*. [10], [11], Hernandez *et al*. [12], Kokkolaras *et al*. [13], Kumar *et al*. [14], Li and Azarm [15], Messac *et al*. [16], Nelson *et al*. [17], Ortega *et al*. [18], Seepersad *et al*. [19], [20], Simpson *et al*.[21], [22], Willcox and Wakayama [23]). However, other approaches optimize for the platform selection and product family design simultaneously; that is, platforms are specified *a posteriori* (Akundi *et al*. [24], Cetin and Saitou [25], de Weck *et al*. [26], Fellini *et al*., [27], [28], Fujita and Yoshida [29], Gonzales-Zugasti and Otto [30], Hernandez *et al*. [31], [32], Messac *et al*. [33], Nayak *et al*.[34], Rai and Allada [35], Hassan *et al*. [36], Simpson and D'souza [37], Fujita *et al*. [38], Khire and Messac [39], Khajavirad *et al*. [40]). Fujita [41] provides a related classification by defining three classes of product family optimization problems: In class-I problems, product attributes are optimized under a fixed platform assumption (i.e. the platform is known *a priori*); class-II deals with finding the optimal platform using predefined product attributes; and finally, in class-III, the product attributes and platform are optimized simultaneously. In general, only the class-III *a posteriori* approaches can claim to guarantee optimality with respect to the joint problem in general, since platform selection and variable optimization are not independent, and it would be difficult or impossible in most cases to know the optimal platform without first knowing something about the design variable values at the solution.

1.1 Prior Approaches for Solving the Joint Problem

Simpson *et al*. [1] classify approaches for solving the joint *a posteriori* platform selection and design problem based on the number of stages used for finding the optimal solution: Single-stage approaches optimize both platform variable selection and the design of the family of products simultaneously (Akundi *et al*. [24]; Cetin and Saitou [25], Fujita *et al.* [38], Fujita and Yoshida [29], Gonzales-Zugasti and Otto [30], Hassan *et al.* [36], Simpson and D'souza [37], Khire and Messac [39], Khajavirad *et al*. [40]), whereas two or multi-stage algorithms select the platform within the first stage and fix the selection while optimizing the product family design in the second stage (de Weck *et al.* [26], Hernandez *et al*. [31], [32], Messac *et al*., [33], Nayak *et al*. [34], Fellini *et al.* [27], [28], Rai and Allada[35]). There is some tradeoff between single and two-stage approaches: Optimizing the platform and corresponding design variables in two separate stages may lead to sub-optimal solutions. However, single stage approaches tend to have higher computational cost, which can make these algorithms impractical when large numbers of products are considered. To sum up, a main challenge in product family optimization is to design a single-stage approach that solves the joint problem and remains efficient and scalable while dealing with large problems.

Most prior single-stage approaches use genetic algorithms (GAs) for solving the joint-problem (Simpson and D'souza [37], Hessan *et al*. [36], Akundi *et al*. [24], Gonzales-Zugasti and Otto [30], Cetin and Saitou [25], Khajavirad *et al.* [40]). However, applying stochastic methods like GAs to the joint problem involves significant computational cost and limited scalability for dealing with large problems. Khajavirad *et al*. [40] proposed an innovative decomposition method that significantly improves scalability of the GA approach; however, the reliance of these approaches on GAs limits the ability to ensure local or global optimality and requires significant time in problem-specific algorithm design and parameter tuning.

Therefore, an alternative method that is able to take advantage of the properties of established gradient-based algorithms in solving the joint problem would be beneficial. However, the platform-selection phase involves discrete variables, leading to a mixed integer nonlinear programming (MINLP) formulation, which is challenging to solve directly. Two prior approaches have relaxed the MINLP formulation to the continuous domain using a sequence of approximation functions so that nonlinear programming (NLP) techniques can be used: Fellini *et al* [27] proposed an approach for module-based platforms with generalized commonality and Khire and Messac [39] proposed an approach for scale-based platforms with all-or-none commonality. We discuss each of these approaches in turn.

The approach of Fellini *et al*. [27] involves an approximation of the binary commonality variables using a continuous, differentiable function and defined tolerances for categorizing the shared and distinct variables after optimization in the relaxed space. Although the approach initially searches in the relaxed joint space, it is not a single-stage approach for solving the joint problem: In the first stage, the problem is formulated for maximizing commonality, and the performance objectives are treated as constraints on minimum acceptable deviation. The second stage designates variables as common or distinct based on results from the first stage and holds this designation fixed during variant optimization. Hence, the first step results in finding a feasible platform set, which is not necessarily a unique solution, and that set is optimized for maximum performance in the second stage. This approach has been demonstrated to be efficient for optimizing the product family using gradient-based methods; however, the two-stage approach may lead to sub-optimal solutions. Furthermore, increasing the number of products in the family increases the number of feasible platform alternatives considerably. In this case, even if the first stage can find all feasible platforms, they must all be optimized in the second stage in order to ensure the jointly optimal solution, which makes this method computationally inefficient for a large number of products.

Khire and Messac [39] applied the selection integrated optimization method (SIO), which integrates the platform selection and variant design optimization phases by using a variable segregating mapping function (VSMF). VSMF is defined as a family of continuous functions that progressively approximate the discontinuous mapping used to segregate platform variables from scaled design variables in scale-based product families. Hence, the joint product family optimization problem was formulated as a series of continuous optimization problems so that the final solution defines the platform and non-platform design variables based on a predefined threshold value for each design variable. The proposed method was illustrated in the design of the electric motor product family and proved to be robust for optimizing the joint problem for scale-based product families in a single stage approach. However, the method does not address module-based platforms, and platform selection is limited to the all-or-none sharing possibility.

1.2 Commonality Metrics

Khajavirad and Michalek [42] argue that prior metrics for measuring the degree of commonality in the product family optimization literature do not properly address the producer's purpose for designing product families. Some metrics penalize increased variance among variables [24], [33], [36], [37], [39],, whereas it is often a binary "common or not common" decision that determines the ability to share use of tooling and equipment, thus increasing economies of scale and reducing cost. Some methods restrict commonality to all-or-none, eliminating the possibility of component sharing among a subset of the variants [24], [33], [36], [37], [39], and other metrics double-count commonality [27]. To address these issues Khajavirad and Michalek [42] proposed a new method for computing the Commonality Index (CI), introduced by Martin and Ishii [43], using information available during the optimization process, and they relaxed the discrete formulation, verifying the validity of the proposed metric in the continuous space and comparing properties to prior metrics. The resulting continuous CI metric, along with the proposed consistency relaxation function, is adopted here for optimizing the joint problem using NLP techniques.

1.3 Scalability and Decomposition

Although using gradient-based methods decreases computational cost considerably, a gradient-based algorithm may still encounter scalability limitations when dealing with large number of products. One systematic way to handle large-scale optimization problems with special structures is to decompose the original problem hierarchically into a number of smaller sub-systems that are optimized separately and coordinated to arrive at the overall system optimum. While a variety of decomposition methods have been studied in the literature for decomposing complex systems, analytical target cascading (ATC) was developed specifically for solving a hierarchy of interacting systems and subsystems, and prior research has applied the framework to product line and product family optimization (Kokkolaras *et al*. [13], Michalek *et al*. [44]). Convergence proofs are available for ATC, and ATC avoids the numerical problems of some alternative decomposition methods (Kim *et al*. [45], Michelena *et al.* [46], Tosserams *et al*.[47], Li *et al*. [48]). Hence, ATC has been widely used for optimizing engineering design problems with hierarchical structures (Kim *et al.* [8], [9] and [49], Kokolaras *et al*. [13], Papalambros [50], Choudhary *et al*.[51], Allison *et al.* [52], Michalek *et al.* [44], [53]). In particular, Kokolaras *et al*. [13] extended the target cascading methodology for optimal product development to the design of product families with predefined platform architecture. According to their framework, the top level problem addresses family attributes while lower levels (i.e. product levels) address the attributes associated with particular components to satisfy individual product requirements. Component sharing is represented by introducing elements with multiple parents. They applied ATC for designing a product family with two vehicles by decomposing it into a four-level vehicle design problem. However, this method is only applicable to product families with fixed (*a priori*) platform configurations. Michalek *et al.* [44] applied ATC to design product lines using market data to predict demand and revenue and manufacturing models to predict cost. This approach went further to quantify cost and revenue benefits of product line decisions; however, the approach did not address commonality among products in the line.

1.4 Proposed Approach

In this paper, we propose a single-stage approach for solving the joint platform-selection and design problem for module-based product families using gradient-based methods. The combinatorial platform variable selection is relaxed to the continuous space using the commonality index extension introduced by Khajavirad and Michalek [42]. Next, in order to make the algorithm scalable, the original all-in-one formulation is decomposed into a two-level optimization problem using ATC so that the system level optimization problem finds the optimal platform configuration while each sub-system deals only with optimization of a single variant in the family. Finally, a case study involving the design of a family of bathroom scales from the literature is presented and optimized using the proposed approach.

2. PROPOSED METHODOLOGY

The proposed methodology is developed by first deriving the original all-in-one formulation and then decomposing the formulation using ATC.

2.1 All-In-One Formulation

The joint product family platform selection and design problem can be considered as an extension of single product design optimization by including a commonality metric as an additional objective (along with the individual performance objectives of each variant) and imposing consistency constraints between the platform-selection and design variables. Fellini *et al*. [27] proposed the following formulation for optimizing a product family with n products[1]:

$$
\begin{aligned}
&\text{Maximize} && \left\{ \mathbf{f}^{i}(\mathbf{x}^{i}), \textstyle\sum_{(p,q)ij} \eta_{pq}^{ij} \right\} && i,j = 1,2,\dots,n\ ,\ \ i<j \\
&\text{with respect to} && \boldsymbol{\eta}, \mathbf{x} && (p,q) \in S_{ij} \\
&\text{subject to} && \mathbf{g}^{i}(\mathbf{x}^{i}) \le \mathbf{0} && \qquad (1) \\
& && \mathbf{h}^{i}(\mathbf{x}^{i}) = \mathbf{0} \\
& && \eta_{pq}^{ij}(x_p^i - x_q^j) = 0 \\
& && \eta_{pq}^{ij} \in \{0,1\}
\end{aligned}
$$

where the set S_{ij} contains index pairs of components in products i and j that are *candidates* for sharing, and η^{ij}_{pq} is the commonality decision variable, which remains consistent with its corresponding design variables by imposing the last equality constraint in Eq.(1). Khajavirad and Michalek [42], argued that defining the commonality metric as the sum of the commonality decision variables leads to a "*double counting*" defect, which causes a convergence bias toward product family architectures with all-or-none component sharing. They addressed this problem by reformulating the commonality index (CI) introduced by Martin and Ishii [43], as a function of the commonality decision variables, so that it can be applied as the commonality metric within an optimization context. They also noted that the consistency constraint in Eq(1) does not ensure that the commonality decision variable will equal one at feasible points where the corresponding design variables are equal, and they defined the commonality variables as a function of the corresponding design variables instead of treating them as variables, thus converting a MINLP formulation into a NLP formulation with a discontinuous commonality function. Finally, to simplify notation the problem is restricted[2] such that components are indexed $k = \{1,2,\dots,m\}$ and the set of candidate components S_{ij} consists only of all sets where i=j. Additionally, we no longer assume, as in Eq(1), that each component has a single associated variable. Instead, we explicitly define a vector of variables $\mathbf{x}_k$ for each component (or module) k and enforce that all dimensions of the component must be shared before the component can be considered common. Applying these modifications to Eq.(1), the optimization problem for a family of n products, each with m components, can be reformulated as follow:

$$
\begin{aligned}
&\text{Maximize} && \mathbf{f}^{i}(\mathbf{x}^{i}), && i = 1,\dots,n \\
&\text{Maximize} && CI(\boldsymbol{\Gamma}_k), && k = 1,\dots,m \\
&\text{with respect to} && \mathbf{x} = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^n\}
\end{aligned}
$$

$$
\boldsymbol{\Gamma}_k = \begin{bmatrix} 1 & \eta_k^{12} & \cdots & \eta_k^{1n} \\ \eta_k^{21} & 1 & & \eta_k^{2n} \\ \vdots & & \ddots & \vdots \\ \eta_k^{n1} & \eta_k^{n2} & \cdots & 1 \end{bmatrix}, \quad \eta_k^{ij} = \begin{cases} 1 & \text{If } \mathbf{x}_k^i = \mathbf{x}_k^j \\ 0 & \text{Otherwise} \end{cases}, \quad i,j = 1,\dots,n \qquad (2)
$$

$$
\begin{aligned}
&\text{subject to} && \mathbf{g}^{i}(\mathbf{x}^{i}) \le \mathbf{0} \\
& && \mathbf{h}^{i}(\mathbf{x}^{i}) = \mathbf{0}
\end{aligned}
$$

where $\boldsymbol{\Gamma}_k$ represents the commonality matrix for the k^{th} component in the family. CI is the commonality index introduced by Martin and Ishii [43] as a measure of unique parts: For a product family with a given platform configuration, the commonality level can be calculated as:

$$
CI = 1 - \frac{u - \max m_i}{\sum_{i=1}^{n} m_i - \max m_i} \qquad (3)
$$

where u is the total number of distinct components, m_i represents the number of components used in variant i, and n shows the number of variants in the family. Khajavirad and Michalek [42] reformulated Eq. (3) so that it can be calculated given the available data during the optimization process[3]:

$$
CI = \frac{\sum_{k=1}^{m}\sum_{r=1}^{s_k} (\lambda_r^k - 1)}{m \times (n-1)} \qquad (4)
$$

where $\boldsymbol{\lambda}^k$ is the vector of non-zero eigenvalues of $\boldsymbol{\Gamma}_k$ and s_k is the number of blocks in $\boldsymbol{\Gamma}_k$. In order to solve Eq.(2) using gradient-based methods, both commonality metric and commonality decision variable definitions must be relaxed to the continuous space. Khajavirad and Michalek [42] showed that Eq.(4) can be relaxed to the continuous space using the following modification:

$$
CI = \frac{\sum_{k=1}^{m}\sum_{r=1}^{s_k} (\max(\tilde{\lambda}_r^k, 1) - 1)}{m \times (n-1)} \qquad (5)
$$

where $\tilde{\lambda}_r^k$ represents the r^{th} eigenvalue of the k^{th} commonality matrix[4]. Commonality variables η can be relaxed to the continuous space using various representations. Fellini *et al.* [27] proposed the following approximating function:

$$
\eta_k'^{ij} = \left(1 + \left(\frac{x_k^i - x_k^j}{\alpha} \right)^2 \right)^{-1} \qquad (6)
$$

[1] While there are a number of different formulations for the product family optimization problem, the authors found Fellini's formulation a proper form for representing the joint platform selection and design problem as a MINLP problem.

[2] The restriction does not eliminate the possibility that some variants may not include all modules, since a variant that does not include a particular module can be represented by enforcing η=0. However, the restriction does disallow cases where one variant carries two instances of a module. Extension to include this case is relatively straightforward.

[3] In this case, we assume that the benefit of component sharing is equal across components. If the cost savings associated with each commonality alternative are known, they can be included in a straightforward way.

[4] Using the **max** function for relaxing the discrete definition, introduces a discontinuity in the derivative of CI, which can be eliminated by applying a slack variable. However, our empirical examples indicate the gradient-based algorithms perform well with the formulation in Eq(5).

where α is a value between 0 and 1 that controls the degree to which the curve approximates the discontinuous step function that would describe the discrete nature of commonality: As the α value decreases, the function tends toward that of the discrete formulation. Hence, the discrete optimization problem can be replaced by a series of continuous problems in which α decreases iteratively until variables that are designated as common fall within an acceptable deviation tolerance. However, Khajavirad and Michalek [42] showed that the Hubbert function has better properties for optimization, including improved curve behavior with decreasing α, derivative continuity, and concavity:

$$\eta_k'^{ij} = \frac{4\exp\left(-\Delta x_k / \alpha\right)}{\left(1+\exp\left(-\Delta x_k / \alpha\right)\right)^2}, \qquad \Delta x_k = x_k^i - x_k^j \tag{7}$$

Therefore, using the above modifications and generalizing to multiple variables per component, the NLP formulation for the joint product family platform-selection and design problem is as follows:

$$\begin{aligned}
&\text{Maximize} && \mathbf{f}^i(\mathbf{x}^i), && i = 1, \ldots, n\\
&\text{Maximize} && CI(\mathbf{\Gamma}_k, \alpha_{s+1}), && \alpha_{s+1} = c\alpha_s, \quad k = 1,\ldots,m\\
& && && 0 < c < 1\\
&\text{with respect to} && \mathbf{x} = \{\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^n}\}\\
&\text{subject to} && \mathbf{g}^i(\mathbf{x}^i) \le \mathbf{0}\\
& && \mathbf{h}^i(\mathbf{x}^i) = \mathbf{0}
\end{aligned} \tag{8}$$

$$\text{where } \mathbf{\Gamma}_k' = \begin{bmatrix} 1 & \eta_k'^{12} & \cdots & \eta_k'^{1n}\\ \eta_k'^{21} & 1 & & \eta_k'^{2n}\\ \vdots & & \ddots & \vdots\\ \eta_k'^{n1} & \eta_k'^{n2} & \cdots & 1 \end{bmatrix}, \; \eta_k'^{ij} = \frac{4\exp\left\|\dfrac{\mathbf{x}_k^i - \mathbf{x}_k^j}{\alpha_{s+1}}\right\|}{\left(1+\exp\left\|\dfrac{\mathbf{x}_k^i - \mathbf{x}_k^j}{\alpha_{s+1}}\right\|\right)^2}$$

Hence, by defining α_0 and c, Eq.(8) will be optimized iteratively until the difference between the common variables fall within the acceptable tolerance. It should be noted that in Eq.(8) we used a linear scheme for decreasing the α value. However, in general, different methods, such as an exponential reduction scheme, can be applied depending on the form of the approximating function. Moreover, appropriate values for c depend on the optimization problem and should be tuned properly for each case (it should be noted that there is an optimum choice for c in any particular problem: Smaller values may cause convergence problems and larger values may induce increased computational effort without any effect on the final solution).

It should be noted that Eq.(8) is a multi-objective optimization problem with $1+\sum_{i=1}^{n} p_i$, $i = 1,\ldots,n$ objective functions, where p_i is the number of objective functions for the i^{th} product. In practice we are interested to determine the Pareto frontier of the commonality value versus total performance loss, i.e. the tradeoff between increasing the commonality and loosing variant performance. Hence, all performance objectives can be grouped into one objective defined as the (normalized and possibly weighted) sum of all performance deviations from their corresponding targets. Using this aggregated performance function, the number of objectives in Eq.(8) reduces to two. Moreover, in the discrete definition of commonality variables, for a product family with n products, each with m components, CI can attain the following values:

$$CI = \frac{r}{m\times(n-1)}, \qquad r = 0, \ldots, m\times(n-1) \tag{9}$$

Hence, the Pareto frontier representing the tradeoff between performance loss and commonality can be found by minimizing the performance loss and the commonality deviation with respect to each level given by Eq.(9). Specifically, the multi objective optimization problem is converted to a series of $1 + m(n\text{-}1)$ single objective optimization problems, each finding the optimal platform and individual product design variables for a fixed commonality level. Applying the above modifications, Eq.(8) can be reformulated as follows:

$$\begin{aligned}
&\text{Minimize} && \left(l^r - CI(\mathbf{\Gamma}_k, \alpha_{s+1})\right)^2 + \sum_{i=1}^{n} w_i \left\|\mathbf{T}^i - \mathbf{f}^i(\mathbf{x}^i)\right\|_2^2\\
& && k = 1,\ldots,m\\
& && \alpha_{s+1} = c\alpha_s, \qquad 0 < c < 1\\
& && l^r = \frac{r}{m\times(n-1)}, \quad r = 0,\ldots,m\times(n-1)\\
&\text{with respect to} && \mathbf{x} = \{\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^n}\}\\
&\text{subject to} && \mathbf{g}^i(\mathbf{x}^i) \le \mathbf{0}\\
& && \mathbf{h}^i(\mathbf{x}^i) = \mathbf{0}
\end{aligned} \tag{10}$$

$$\text{where } \mathbf{\Gamma}_k' = \begin{bmatrix} 1 & \eta_k'^{12} & \cdots & \eta_k'^{1n}\\ \eta_k'^{21} & 1 & & \eta_k'^{2n}\\ \vdots & & \ddots & \vdots\\ \eta_k'^{n1} & \eta_k'^{n2} & \cdots & 1 \end{bmatrix}, \; \eta_k'^{ij} = \frac{4\exp\left\|\dfrac{\mathbf{x}_k^i - \mathbf{x}_k^j}{\alpha_{s+1}}\right\|}{\left(1+\exp\left\|\dfrac{\mathbf{x}_k^i - \mathbf{x}_k^j}{\alpha_{s+1}}\right\|\right)^2}$$

$\mathbf{T}^i$ represents the vector of the performance targets for the i^{th} product, and l^r shows the commonality target value for $nm\text{–}r$ distinct components. The w^i terms are weighting coefficients that define the relative importance of achieving each performance objective.

2.2 Decomposed Formulation

According to the ATC framework, the original all-in-one problem with a hierarchical structure is decomposed into a top level supersystem and a hierarchy sub-systems. The overall system objective function is the sum of all of the objective functions presented in each sub-problem, and subproblems are defined so that they are nearly separable expect for a few variables called linking variables[5]. The top-element, which

[5] In the ATC literature, the term "linking variable" is sometimes used to refer only to variables shared between subsystems at the same level of the hierarchy; however, in the general decomposition literature, the term is used to

represents the overall system, propagates deign targets to the subsystems below. Each subproblem is optimized separately to meet its targets as closely as possible. Then lower level systems pass up responses, which are rebalanced at higher levels iteratively until consistency is achieved.

As can be seen in Eq.(10), the commonality deviation portion of the objective function is the only non-separable part in the all-in-one formulation. If commonality is not considered, each product could be optimized independently. Hence, using ATC, the joint product family platform-selection and design problem can be decomposed to a two-level optimization problem: The system level optimization problem finds the optimal platform configuration while each subsystem only deals with optimizing a single product in the family (Figure 1).

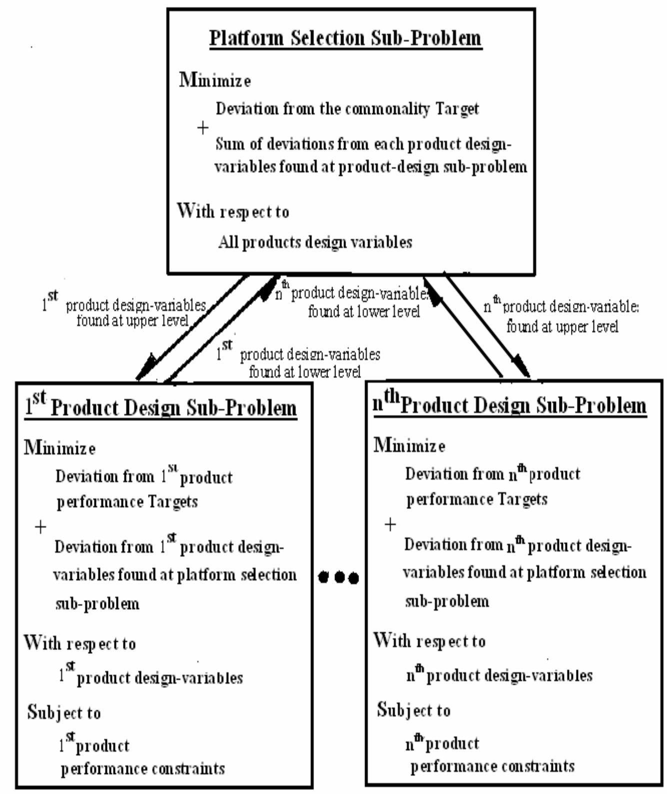

Figure 1. ATC framework for optimizing the joint product family problem

The resulting system level problem is an unconstrained NLP problem, which finds the optimal platform and distinct design variables for a given number of shared components defined by the target value and with minimum deviation from the responses passed up from product level sub-problems:

$$\text{Minimize} \quad \left(l^r - CI(\boldsymbol{\Gamma}_k, \alpha_{s+1})\right)^2 + \sum_{i=1}^{n} \pi(\mathbf{x}^i - \mathbf{y}^i)$$

$$k = 1,..,m \quad , \quad \alpha_{s+1} = c\alpha_s , \quad 0 < c < 1$$

$$l^r = \frac{r}{m\times(n-1)}, \qquad r =\in \{0,...,m\times(n-1)\}$$

$$\text{with respect to} \quad \mathbf{x} = \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^n\} \qquad (11)$$

$$\boldsymbol{\Gamma}'_k = \begin{bmatrix} 1 & \eta_k'^{12} & \cdots & \eta_k'^{1n} \\ \eta_k'^{21} & 1 & & \eta_k'^{2n} \\ \vdots & & \ddots & \vdots \\ \eta_k'^{n1} & \eta_k'^{n2} & \cdots & 1 \end{bmatrix}, \; \eta_k'^{ij} = \frac{4\exp\left\|\dfrac{\mathbf{x}_k^i - \mathbf{x}_k^j}{\alpha_{s+1}}\right\|}{\left(1+\exp\left\|\dfrac{\mathbf{x}_k^i - \mathbf{x}_k^j}{\alpha_{s+1}}\right\|\right)^2}$$

in which π is the inconsistency constraint relaxation function, which forces the response copies to match the targets. $\mathbf{y}^i$ represents the response vector (i.e. the product design variables) passed up from the i^{th} sub problem.

The optimization problem for the i^{th} subsystem, which optimizes the i^{th} individual product has the following formulation:

$$\text{Minimize} \qquad w_i \left\|\mathbf{T}^i - \mathbf{f}^i(\mathbf{y}^i)\right\|_2^2 + \pi(\mathbf{y}^i - \mathbf{x}^i) \qquad (12)$$

$$\text{with respect to} \quad \mathbf{y}^i$$

$$\text{subject to} \quad \mathbf{g}^i(\mathbf{y}^i) \le \mathbf{0}$$

$$\mathbf{h}^i(\mathbf{y}^i) = \mathbf{0}$$

In which $\mathbf{y}^i$ is the vector of local variables representing the design variables for the corresponding product, and $\mathbf{x}^i$ is the target vector cascaded down from the system level subproblem.

A variety of approaches have been used to decompose and coordinate consistency among subsystems, including quadratic penalty functions (Kim *et al.* [45], Michelena *et al.* [46], Michalek and Papalambros [54]), ordinary Lagrangian relaxation (Lassiter *et al.* [55]), and augmented Lagrangian relaxation (Tosserams *et al.* [47], Kim *et al.* [45], Li *et al.* [48]). A recent comparison study by Li *et al.* [48] concluded that the truncated approaches of the augmented Lagrangian alternating direction (ALAD) method of multipliers (Tosserams *et al.* [47], [56]) and the diagonal quadratic approximation (DQA) approach (Li *et al.* [48]) have the best computational efficiencyby orders of magnitude in empirical examples, and we adopt the ALAD method in this study. According to this method, the consistency constraint relaxation function $\pi(\mathbf{y}^i - \mathbf{x}_m^i)$ is the augmented Lagrangian function $\boldsymbol{\lambda}^T(\mathbf{y}^i - \mathbf{x}_m^i) + \mathbf{w}^T(\mathbf{y}^i - \mathbf{x}_m^i)^2$, where the values of $\boldsymbol{\lambda}$ are determined using the method of multiplers. In the ALAD approach, each sub-problem is solved only once before updating $\boldsymbol{\lambda}$ via the method of multipliers, instead of solving the iterative inner loop coordination scheme to optimality for each fixed value of $\boldsymbol{\lambda}$, as required by the standard augmented Lagrangian method. The penalty weight may be held constant or only be updated when no improvement in objective function is observed.

The optimization algorithm for solving the decomposed product family problem is sketched in Figure 2. As can be

refer to variables shared between any two subsystems. Here we use the more general definition.

observed from this figure, in the inner loop, for a fixed α and commonality value, the ATC formulation is solved using ALAD method: Langrange multipliers $\boldsymbol{\lambda}$ are updated according to the method of multipliers and w^i is only updated if no improvement in its corresponding objective function was observed. This process continues until the value of the inconsistency constraint in all subproblems falls below the maximum allowable deviation defined by the designer. In the middle loop, α decreases iteratively until the deviation of the commonality metric from its target value and the relative difference among the shared components falls below user specified tolerances. Next, after, finding the optimal platform configuration and individual products for a constant commonality value, l^r is incremented, and this procedure continues until the optimum product families for the entire range of commonality levels are found.

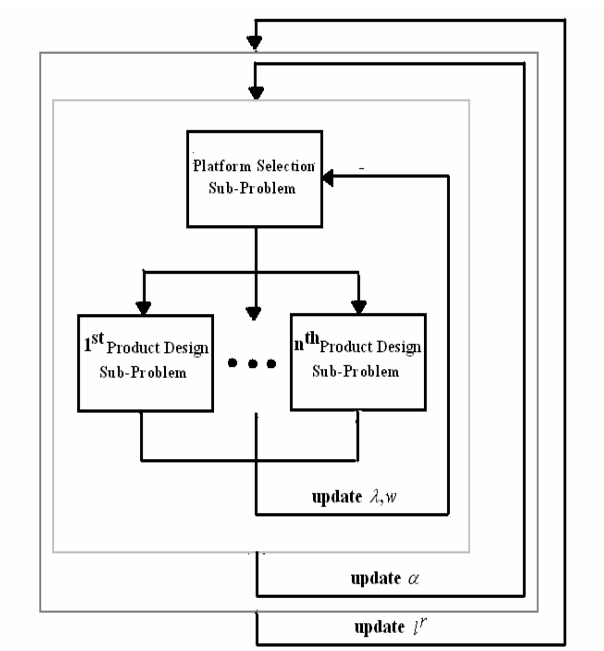

Figure 2. Decomposition algorithm for optimizing the joint product family problem

3. CASE STUDY: BATHROOM SCALE DESIGN

We now apply the proposed approach for optimizing the joint product family problem to the design of a family of standard household dial-read out and digital bathroom scales from the literature in order to illustrate the approach and examine efficiency. Design of a family of scales is a well-suited example for illustrating the trade off between commonality and achievement of distinct performance targets[6] because individual products with distinct characteristics and performance objectives (e.g. digital and analog) operate according to nearly identical principles: The force applied on the top cover B is amplified by four levers A that transfer the force to a coil spring C at the base of the scale (Figure 3). The spring resists displacement proportionally to the force applied, and a pivot D transfers motion to a horizontal rack E, which turns a pinion gear F attached to the dial G. The result is dial turn per force applied. In the analogue case (Figure 4a) the dial is read directly by the user. In the digital case (Figure 4b) the dial is an encoder wheel, which is read by a photointerrupter and displayed on the digital display. The potential for achieving significant market differentiation with a high level of engineering commonality makes the case study well-suited to product family optimization.

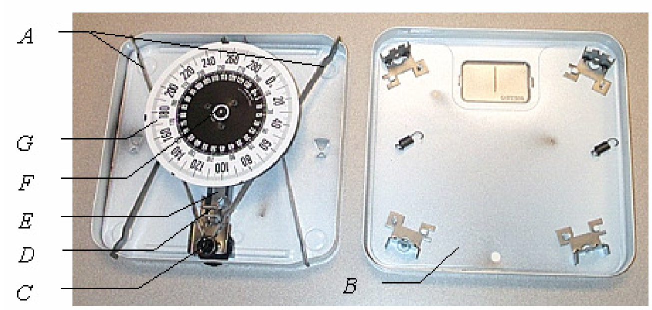

Figure 3. Disassembled analog scale showing components

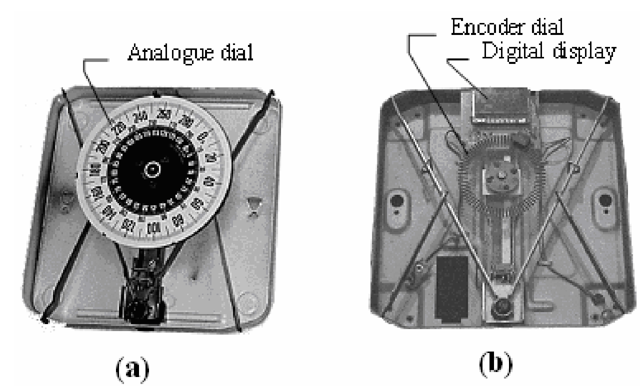

Figure 4. Scale with cover removed a) dial read-out scale, b) digital scale

The engineering model used for optimizing the bathroom scale family is taken from Michalek *et al.* [53]. Product design variables are depicted in Figure 5 on the analog scale, and the digital scale has the same design variables except for the dial diameter, which is not present in the digital scale. A brief description of design variables, bounds, and fixed parameters is provided in Appendix A.

[6] Following the bulk of the product family literature, we have treated performance targets as exogenous and introduced a generic penalty function for deviation from those targets. If data are available, quantification of differentiation in terms of the market responses of a heterogeneous consumer population would more completely describe the product family tradeoff [44]; however, we do not pursue this here.

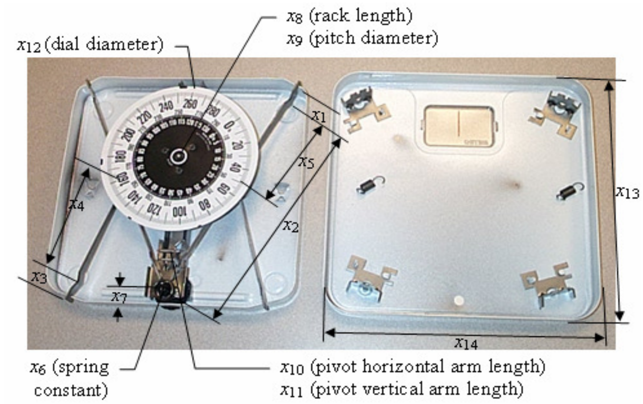

Figure 5. Design variables shown on the disassembled analog scale (Michalek *et al.* [53])

For module-based product family optimization, design variables are grouped according to the component to which they belong. These components are depicted on the analog scale in Figure 3 and listed in Table 1. It should be noted that in platform based product families, which is the focus of this paper, commonality is measured based on component sharing; i.e. two products have a common part if *all* of the corresponding design variables have the same value for both products.

Table 1: Scale components and their design variables

	Component Name	Associated Variables
1	Long Lever (A)	$\{x_1, x_2, x_5\}$
2	Cover (B)	$\{x_7, x_{13}, x_{14}\}$
3	Spring (C)	$\{x_6\}$
4	Pivot (D)	$\{x_{10}, x_{11}\}$
5	Short Lever (A)	$\{x_3, x_4\}$
6	Rack (E) & Pinion (F)	$\{x_8, x_9\}$
7	Dial (G)	$\{x_{12}\}$

Performance objectives are the same as those addressed by Michalek *et al.* [53], except for the tick mark gap, which is not considered in this study (Table 2).

Table 2: Performance Objectives for Scale Design

Product Characteristic	Formula
Weight Capacity	$z_1 = \dfrac{4\pi x_6 x_9 x_{10}(x_1 + x_2)(x_3 + x_4)}{x_{11}(x_1(x_3 + x_4) + x_3(x_1 + x_5))}$
Aspect Ratio (Analog Scale)	$z_2 = \dfrac{x_{13}}{x_{14}}$
Platform Area	$z_3 = x_{13}x_{14}$
Number Size (Analog Scale)	$z_4 = \dfrac{\left(2\tan\left(\pi y_8 z_1^{-1}\right)\right)\left(\frac{1}{2}x_{12} - y_7\right)}{\left(1 + 2y_3^{-1}\tan\left(\pi y_8 z_1^{-1}\right)\right)}$

Design constraints are detailed in Appendix A: In addition to the constraints developed in [53], we added additional constraints to the optimization problem in order to capture additional design issues ignored in the prior study.

4. NUMERICAL RESULTS:

In order to illustrate the concept of generalized component sharing in the optimal product family, the proposed approach is first applied for solving a family of three bathroom scales; including one analog and two digital scales. The commonality metric for a platform-based product family measures the number of shared/distinct *components* (not design variables), so Δx_k in Eq.(6) was generalized in Eq.(7) using a norm of the vector of deviations for the component. In the application we choose the normalized l_1 norm and divide by the number of variables to measure the average deviation in each component. For example, $\Delta \boldsymbol{x}_1$ is defined as follows:

$$\Delta \mathbf{x}_1 = \frac{1}{3}\left(\left|\frac{x_1^1 - x_1^2}{x_{1\max} - x_{1\min}}\right| + \left|\frac{x_2^1 - x_2^2}{x_{2\max} - x_{2\min}}\right| + \left|\frac{x_5^1 - x_5^2}{x_{5\max} - x_{5\min}}\right|\right) \qquad (12)$$

where x_{imax} and x_{imin} represent the upper and lower bounds for the design variables respectively. Performance targets for individual products are listed in Table 3. Performance targets were picked from the product attribute levels suggested by Michalek *et al.* [53] and checked for feasibility; i.e. they should be fully achieved under the no-commonality condition so that any performance loss in the family shows the component-sharing effect. Furthermore, targets were chosen so that the performance characteristics of each individual product become distinct from those of others to have the least amount of component sharing when individual products are optimized independently.

Table 3: Performance Targets for the family of the three bathroom scales

Product Attribute	Analog Scale	1st Digital Scale	2nd Digital Scale
Weight Capacity	300	320	280
Aspect Ratio	1.0	0.80	1.20
Platform Area	130	140	120
Number Size	1.2	----	---

Since in the digital scale number size is not a function of design variables considered in this study, it is not treated as a performance objective for the 2nd and 3rd products. Before applying ATC for optimizing the joint product family problem, each individual product is optimized separately to find the best achievable performance under the zero-enforced-commonality condition (i.e. components that are shared among the products by chance, without imposing any commonality constraint or objective). Optimal designs are listed in Table 4; both the analog scale and the 2nd digital scale have the same rack and pinion. Hence, the minimum commonality value is 1/12 from Eq.(8) and the product family should be optimized for l_r =2/12,…, 1.

Next, the ATC framework described in the previous section (Figure 2) is applied for optimizing the family of three bathroom scales. The performance objective is the average normalized deviation of all product attributes from their corresponding targets over the entire family. Results are listed in Appendix B for CI from 2/12 (minimum commonality) to 1 (complete commonality). The Pareto curve is sketched in Figure 6. The Pareto curve shows the trade-off between the commonality and the ability of the variants to achieve their distinct performance targets. As can be seen from Figure 6, up to a level of around 50% commonality the rate of performance loss due to increasing commonality is small; that is we can decrease the manufacturing cost considerably while maintaining individual distinctiveness.

Table 4. Optimal products for the minimum CI

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.53 9.21 2.61	2.52 8.63 2.61	1.74 9.95 2.37
Cover	11.53 11.38 0.50	10.58 13.23 0.55	12.00 10.00 0.70
Spring	139.99	160.00	96.83
Pivot	0.54 1.89	0.50 1.74	0.50 1.90
Short Lever	3.28 3.38	3.17 3.28	3.30 4.51
Rack & Pinion	6.80 0.25	6.60 0.25	6.80 0.25
Dial	9.33	---	----

However, by increasing commonality beyond 50%, performance loss grows more rapidly. This effect can be quantified by observing the product attribute values from the solution tables in Appendix B: Up to CI=9/12, increasing commonality only causes small deviation from the associated targets, but after that level, as more components are forced to be common, attribute values for the variants converge, and the family lose its differentiation; that is, the product family fails to offer distinct products for targeting different market segments. The product family Pareto front gives a thorough perspective to the designer on how to decide about the proper commonality level and its corresponding platform configuration to reduce manufacturing cost without excessive sacrifice of distinctiveness.

Furthermore, this test case reveals the importance of considering the general form for the commonality metric in the optimization formulation: Most of the optimal product families in the case study involve commonality among subsets of the family, which is disallowed under most prior all-or-none approaches.

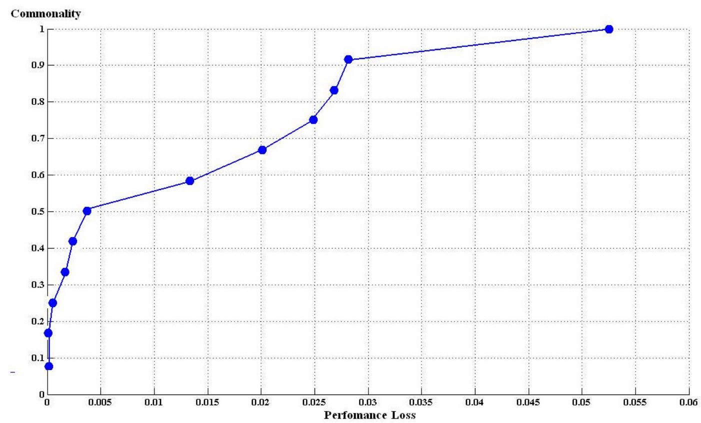

Figure 6. Pareto curve for family of three bathroom scales

Next, in order to show the scalability of the ATC framework, the proposed approach has been applied for optimizing a family of ten scales including five analog and five digital scales. The Pareto curve is sketched in Figure 7. As in the previous case, individual products were first optimized separately to find the minimum commonality value, which for this case is equal to 21/54. Therefore, the product family was optimized for cases l_r=22/54, …, 1. As can be seen from Figure 7, the Pareto curve can be divided into three sections: In the first part, we can increase the commonality up to 55% without any performance loss, which reveals the importance of solving the joint product family problem for cost savings. In the next region, i.e. for 0.55<CI<0.70, the rate of performance loss is slow; that is product attributes deviate only nominally from the assigned targets. However, beyond that limit, the performance loss increases rapidly; that is, individual products cannot achieve their target characteristics.

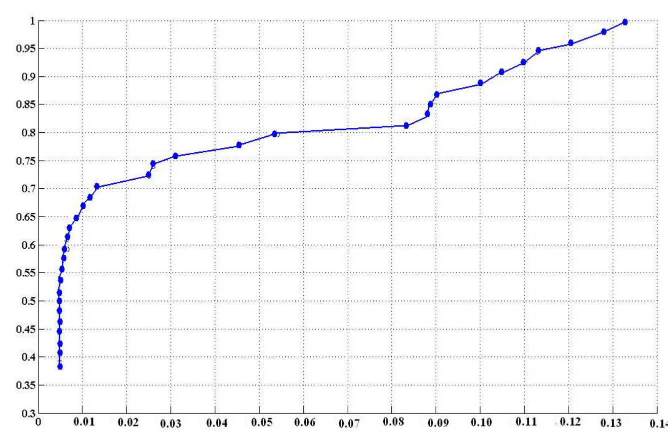

Figure 7. Pareto curve for family of ten bathroom scales

CONCLUSIONS

In this paper, we proposed a novel single-stage approach for optimizing the joint platform selection and product family design problem using gradient-based methods. The commonality metric introduced in the companion paper [42] was used as the commonality objective, and the Hubbert curve was applied for relaxing the binary commonality variables. In order to address the scalability of the proposed method, the all-in-one formulation was decomposed using ATC into a two level optimization problem in which the upper level problem finds the optimal platform configuration while each sub-problem optimizes the individual products. The proposed approach was demonstrated in optimizing families of three and ten bathroom scales. The Pareto optimal fronts for both cases reveal the tradeoff between commonality and the ability to achieve distinct performance targets, which can help in product family planning. Moreover, existence of optimal solutions where components are shared among a subset of the variants points to the importance of applying the generalized commonality metric in the optimization formulation.

In future work, we intend to study efficiency and scalability more closely, comparing the decomposed approach against the all-in-one formulation in terms of computational cost for various numbers of products. In addition, the effect of parallel computing can be examined by applying the DQA relaxation and coordination scheme, which enables use of parallel processing [48]. Finally, the Pareto fronts obtained using both CI and other proposed metrics (including both all-or-none commonality and Fellini's metric) as the commonality objective can be compared, and the trade-off between achieving better solutions using the generalized case vs. the additional computational cost caused by the generalization can be investigated.

ACKNOWLEDGEMENTS

This work is supported in part by the Pennsylvania Infrastructure Technology Alliance, a partnership of Carnegie Mellon, Lehigh University, and the Commonwealth of Pennsylvania's Department of Community and Economic Development (DCED).

REFERENCES

1. Simpson, T. W., 2005, "Methods for Optimizing Product Platforms and Product Families: Overview and Classification," *Product Platform and Product Family Design: Methods and Applications*, T. W. Simpson, Z. Siddique and J. Jiao, eds., Springer, New York, pp. 133-156
2. Allada, V. and Jiang, L., 2002, "New Modules Launch Planning for Evolving Modular Product Families," *ASME Design Engineering Technical Conferences - Design for Manufacturing Conference,* Montreal, Quebec, Canada, ASME, Paper No. DETC2002/DFM-34190.
3. Blackenfelt, M., 2000, "Profit Maximisation while Considering Uncertainty by Balancing Commonality and Variety Using Robust Design - The Redesign of a Family of Lift Tables," *ASME Design Engineering Technical Conferences - Design for Manufacturing,* Baltimore, MD, USA.
4. D'Souza, B. and Simpson, T. W., 2003, "A Genetic Algorithm Based Method for Product Family Design Optimization," *Engineering Optimization,* 35(1), pp. 1-18.
5. Dai, Z. and Scott, M. J., 2004, "Product Platform Design through Sensitivity Analysis and Cluster Analysis," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Salt Lake City, UT, USA.
6. 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.
7. Fellini, R., Papalambros, P. and Weber, T., 2000, "Application of Product Platform Design Process to Automotive Powertrains," *8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* Long Beach, CA, USA.
8. Kim, H.M., D.G. Rideout, P.Y. Papalambros and J.L. Stein, "Analytical Target Cascading in Automotive Vehicle Design," *Proceedings of the 2001 ASME Design Automation Conference*, DAC-21079, Pittsburgh, Pennsylvania, USA.
9. Kim, H, Michelena N.F., Papalambros P.Y., and Jiang, T. "Target cascading in optimal system design," Journal of Mechanical Design, 125(3):474 – 480, September 2003.
10. Gonzalez-Zugasti, J. P., Otto, K. N. and Baker, J. D., 2000, "A Method for Architecting Product Platforms," *Research in Engineering Design,* 12(2), pp. 61-72.
11. Gonzalez-Zugasti, J. P., Otto, K. N. and Baker, J. D., 2001, "Assessing Value for Platformed Product Family Design," *Research in Engineering Design,* 13(1), pp. 30-41.
12. Hernandez, G., Simpson, T. W., Allen, J. K., Bascaran, E., Avila, L. F. and Salinas, F., 2001, "Robust Design of Families of Products with Production Modeling and Evaluation," *ASME Journal of Mechanical Design,* 123(2), pp. 183-190.
13. Kokkolaras, M., Fellini, R., Kim, H. M., Michelena, N. F. and Papalambros, P. Y., "Extension of the target cascading formulation to the design of product families," *Structural and Multidisciplinary Optimization*, Vol. 24, No. 4, pp. 293-301.
14. Kumar, R., Allada, V. and Ramakrishnan, S., 2004, "Ant Colongy Optimization Methods for Product Platform Formation," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Salt Lake City, UT, USA.
15. Li, H. and Azarm, S., 2002, "An Approach for Product Line Design Selection Under Uncertainty and Competition," *ASME Journal of Mechanical Design,* 124(3), pp. 385-392.
16. Messac, A., Martinez, M. P., and Simpson, T. W., 2002, "Effective Product Family Design Using Physical Programming and the Product Platform Concept Exploration Method," *Eng. Optimiz.*, 34, pp. 245–261.
17. Nelson, S. A., II, Parkinson, M. B. and Papalambros, P. Y., 2001, "Multicriteria Optimization in Product Platform Design," *ASME Journal of Mechanical Design,* 123(2), pp. 199-204.
18 Ortega, R., Kalyan-Seshu, U. and Bras, B., 1999, "A Decision Support Model for the Life-Cycle Design of a Family of Oil Filters," *ASME Design Engineering*

Technical Conferences - Design Automation Conference, Las Vegas, NV, ASME, Paper No. DETC99/DAC-8612.

19. Seepersad, C. C., Hernandez, G. and Allen, J. K., 2000, "A Quantitative Approach to Determining Product Platform Extent," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Baltimore, MD, USA.
20. Seepersad, C. C., Mistree, F. and Allen, J. K., 2002, "A Quantitative Approach for Designing Multiple Product Platforms for an Evolving Portfolio of Products," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Montreal, Quebec, Canada,.
21. Simpson, T. W., Maier, J. R. A. and Mistree, F., 1999, "A Product Platform Concept Exploration Method for Product Family Design," *Design Theory and Methodology - DTM'99,* Las Vegas, Nevada, USA.
22. Simpson, T. W., Maier, J. R. A. and Mistree, F., 2001, "Product Platform Design: Method and Application," *Research in Engineering Design,* 13(1), pp. 2-22.
23. Willcox, K, Wakayana, S, "Simultaneous Optimization of a Multiple-Aircraft Family" Journal of Aircraft 2003 0021-8669 vol.40 no.4 (616-622)
24. Akundi, S., Simpson, T. W. and Reed, P. M., 2005, "Multi-objective Design Optimization for Product Platform and Product Family Design Using Genetic Algorithms," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Long Beach, CA, USA.
25. Cetin, O. L. and Saitou, K., 2004, "Decomposition-Based Assembly Synthesis for Structural Modularity," *ASME Journal of Mechanical Design,* 126(2), pp. 234-243.
26. de Weck, O., Suh, E. S. and Chang, D., 2003, "Product Family and Platform Portfolio Optimization," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Chicago, IL, USA.
27. Fellini, R., Kokkolaras, M., Papalambros, P. and Perez-Duarte, A. "Platform Selection under Performance Bounds in Optimal Design of Product Families," Journal of Mechanical Design, July 2005, Vol. 125, No. 4, pp. 524-535.
28. Fellini, R., Kokkolaras, M., Michelena, N., Papalambros, P., Saitou, K., Perez-Duarte, A. and Fenyes, P. A., 2002, "A Sensitivity-Based Commonality Strategy for Family Products of Mild Variation, With Application to Automotive Body Structures," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* Atlanta, GA, USA.
29. Fujita, K. and Yoshida, H., 2001, "Product Variety Optimization: Simultaneous Optimization of Module Combination and Module Attributes," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Pittsburgh, PA, USA.
30. Gonzalez-Zugasti, J. P. and Otto, K. N., 2000, "Modular Platform-Based Product Family Design," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Baltimore, MD, USA.
31. Hernandez, G., Allen, J. K. and Mistree, F., 2002, "Design of Hierarchic Platforms for Customizable Products," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Montreal, Quebec, Canada.
32. Hernandez, G., Allen, J. K. and Mistree, F., 2003, "Platform Design for Customizable Products as a Problem of Access in a Geometric Space," *Engineering Optimization,* 35(3), pp. 229-254.
33. Messac, A., Martinez, M. P. and Simpson, T. W., 2002, "A Penalty Function for Product Family Design Using Physical Programming," *ASME Journal of Mechanical Design,* 124(2), pp. 164-172.
34. Nayak, R. U., Chen, W., and Simpson, T. W., 2002, "A Variation-Based Methodology for Product Family Design," Eng. Optimiz., 34, pp. 69–81.
35. 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. 4075-4098.
36. Hassan, R., de Weck, O. and Springmann, P., 2004, "Architecting a Communication Satellite Product Line," *22nd AIAA International Communications Satellite Systems Conference*
37. Simpson, T. W., and D'Souza, B., 2004, "Assessing Variable Levels of Platform Commonality Within a Product Family Using a Multi-objective Genetic Algorithm," Concurrent Engineering: Research Applications, 12, pp. 119–130.
38. Fujita, K., Akagi, S., Yoneda, T. and Ishikawa, M., 1998, "Simultaneous Optimization of Product Family Sharing System Structure and Configuration," *ASME Design Engineering Technical Conferences - Design for Manufacturing,* Atlanta, GA, USA.
39. Khire, R. A., Messac, A., 2006, "Optimal Design of Product Families using Selection-Integrated Optimization (SIO) Methodology", 11th AIAA/ISSMO Multidisciplinary Analysis and Optmization Conference, Postmouth, Virginia.
40. Khajavirad, A., J. Michalek and T. Simpson (2007) "An Efficient Decomposed Genetic Algorithm for Solving the Optimal Joint Product Platform Configuration and Product Family Design Problem," to appear, *Proceedings of the 3rd AIAA Multidisciplinar Design Optimization Specialists Conference*, April 23-26, Honolulu, Hawaii, USA.
41. Fujita, K, "Product Variety Optimization under Modular Architecture," Computer Aided Design, No. 34, 953-965.
42. Khajavirad, A. and J. Michalek (2007) "An Extension of the Commonality Index for Product Family Optimization," in review, *ASME International Design Engineering Technical Conferences*, September 4-7, Las Vegas, Nevada, USA.
43. Martin, M. and Ishii, K., 1996, August 18-22, "Design for Variety: A Methodology for Understanding the Costs of Product Proliferation," Design Theory and Methodology – DTM'96 (Wood, K., ed.), Irvine, CA, USA.
44. Michalek, J.J., Ceryan, O., Papalambros, P.Y., and Koren, Y., "Balancing marketability and manufacturability in

product line design decision-making". *ASME Journal of Mechanical Design*, 2006.

45. Kim, H. M., Chen, W., and Wiecek, M., "Lagrangian Coordination for Enhancing the Convergence of Analytical Target Cascading," AIAA Journal, Vol. 44. No. 10, pp. 2197-2207, 2006.
46. Michelena, N., Park H., Papalambros, P. Y., "Convergence properties of analytical target cascading". AIAA Journal, 41(5):897 – 905, 2003.
47. Tosserams S., Etman, L.F.P., and Rooda, J.E.. "An augmented Lagrangian relaxation for analytical target cascading using the alternating directions method of multipliers". Structural and Multidisciplinary Optimization, 31(3):176 – 189, 2006.
48. Li, Y., Z. Lu and J. Michalek "Diagonal quadratic approximation for parallelization of analytical target cascading," in review, *ASME Journal of Mechanical Design*.
50. Papalambros, P.Y., "Analytical Target Cascading in Product Development," *Proceedings of the 3rd ASMO UK / ISSMO Conference on Engineering Design Optimization* (O.M. Querin, Ed.), Harrogate, July 9-10, 2001, pp. 3-16. Keynote Presentation.
51. Choudhary, R., Malkawi, A., and Papalambros, P. Y., "Analytic target cascading in simulation-based building design," *Automation in Construction*, Vol. 14, No. 4, 2005, pp. 551-568.
52. Allison, J., Walsh, D., Kokkolaras, M., Papalambros, P. Y., and Cartmell, M., "Analytical Target Cascading in Aircraft Design," *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 9 - 12, 2006.
53. Michalek, J., Feinberg, M., Papalambros, P., "Linking Marketing and Engineering Product Design Decisions via Analytical Target Cascading," *Journal of Product Innovation Management*, 2005, No. 22, pp.42-62.
54. Michalek, J.J., and Papalambros, P.Y., 2005, "An efficient weighting update method to achieve acceptable inconsistency deviation in analytical target cascading," Journal of Mechanical Design, 127(3):206 – 214.
55. Lassiter, J.B., Wiecek, M.M., and Andrighetti, K.R. "Lagrangian coordination and analytical target cascading: Solving ATC-decomposed problems with lagrangian duality," *Optimization and Engineering*, 6(3):361 – 381, September 2005.
56. Tosserams, S., Etman, P., and Rooda, J., "An augmented lagrangian decomposition method for dual block-angular problems in MDO," In Proceedings of the AIAA, 2006

APPENDIX A: ENGINEERING MODEL

Table A1. Engineering Model Design Variables

Design Variable	Lower Bound	Upper Bound
x_1: Length from base to force on long lever	0.125	36.0
x_2: Length from force to spring on long lever	0.125	36.0
x_3: Length from base to force on short lever	0.125	24.0
x_4: Length from force to joint on short lever	0.125	24.0
x_5: Length from force to joint on long lever		
x_6: Spring constant	1.00	200.0
x_7: Distance from base edge to spring	0.50	12.0
x_8: Length of rack	1.00	36.0
x_9: Pitch diameter of pinion	0.25	24.0
x_{10}: Length of pivot's horizontal arm	0.50	1.90
x_{11}: Length of pivot's vertical arm	0.50	1.90
x_{12}: Dial diameter	1.00	36.0
x_{13}: Cover length	1.00	36.0
x_{14}: Cover width	1.00	36.0

Table A2. Engineering Design Model Parameters

Parameter	Value
y_1: Gap between Base and Cover (in)	0.30
y_2: Horizontal Distance between Spring and Pivot	1.10
y_3: Aspect Ratio of Number (Length/Width)	1.29
y_4: Minimum distance from Centerline to Long Lever at Base	2.0
y_5: Minimum distance from Centerline to Short Lever at Base	2.0
y_6: Maximum displacement of spring plate	0.50
y_7: Minimum Distance of support positions from the centerline	0.20
y_8: Number of lbs that Number Length Spans	16.0
y_9: platform area lower bound	100
y_{10}: platform area upper bound	150

Table A3. Geometric Constraints for the Design Problem (Michalek *et al.* [53], ** new constraints)

Constraint Definition	Formula
1. Dial should be small enough to fit in the Analog scale	$x_{12} \le x_{14} - 2y_1$; $x_{12} \le x_{13} - 2y_1 - x_7 - y_9$
2.Joint position of the long and shorts levers should be within the bounds of the long ones	$x_5 \le x_2$
3. Rack must fit inside the scale in the fully extended position	$x_7 + y_9 + x_{11} + x_8 \le x_{13} - 2y_1$
4. Rack must be long enough to span from the pivot to pinion in the analog scale	$x_8 \ge (x_{13} - 2y_1) - (0.5x_{12} + y_1) - x_7 - y_2 - x_{10}$
5. Long levers must fit in the scale within the allowable bounds	$x_1 + x_2 \le \sqrt{(x_{13} - 2y_1 - x_7)^2 + (.5x_{14} - y_1)^2}$ $x_1 + x_2 \ge \sqrt{(x_{13} - 2y_1 - x_7)^2 + (.5x_{14} - y_1 - y_4)^2}$
6. Platform area should remain within the specified range	$y_9 \le x_{13} \times x_{14} \le y_{10}$
7. Maximum displacement of the spring must remain below the allowable value**	$2\pi x_9 x_{10} x_{11}^{-1} \le y_6$
8. Spring applied load should remain the same regardless of legs positions **	$x_1(x_3 + x_4) = x_3(x_1 + x_5)$
9. Short levers should be constrained so that they fit in the scale within the allowable bounds relative to the long levers position **	$x_3 + x_4 \le f_1(x_1, x_2, x_5, x_7, x_{13})$ $x_3 + x_4 \ge f_2(x_1, x_2, x_3, x_{13}, x_{14})$
10. The angular terms should remain within the feasible range	$-1 \le \frac{x_{13} - 2y_1 - x_7}{x_1 + x_2} \le 1$
10. The dial diameter should be restricted so that the dial does not reach to the support position on both levers **	$f_1(x_1, x_2, x_7, x_{12}, x_{13}) \ge 0.5x_{12}$ $f_2(x_1, x_2, x_4, x_5, x_{12}, x_{13}) \ge 0.5x_{12}$
11. The distance from support positions to centerline is constrained to be more than the leg distance from the center line. **	$f_3(x_1, x_2, x_7, x_{13}) \ge y_7$ $f_4(x_3, x_4, x_7, x_8, x_{10}) \ge y_7$

All physical and geometric constraints for the bathroom scale design are listed in Table A3. six new constraints are added to the model proposed by Michalek *et al*. [53] in order to capture aspects of the design that were ignored in the prior modeling. Derivation of the new constraints is presented in the following section:

1. Maximum displacement of the spring must remain below the allowable value: spring displacement is restricted by the scale thickness; hence, the y_6 is introduced as the maximum allowable displacement of the spring:

$$2\pi x_9 x_{10} x_{11}^{-1} \le y_6 \tag{1A}$$

2. The scale should be designed so that it measures the right weight regardless of the consumer legs position. Hence, the applied force on the spring for the most general case (i.e. four different support loads, p_1) should be equal to the one with symmetry assumption (four equal support loads, p_2).

$$p_1 = \frac{1}{4}\left(\frac{2x_1}{(x_1+x_2)} + \frac{2x_3(x_1+x_5)}{(x_1+x_2)(x_3+x_4)}\right) \tag{2A}$$

$$\begin{cases} p_2 = \dfrac{f_1 x_1}{x_1+x_2} + \dfrac{f_2 x_3(x_1+x_5)}{(x_1+x_2)(x_3+x_4)} + \dfrac{f_3 x_1}{x_1+x_2} + \dfrac{f_4 x_3(x_1+x_5)}{(x_1+x_2)(x_3+x_4)} \\ f_1+f_2+f_3+f_4 = 1 \rightarrow f_1+f_3 = 1-f_2-f_4 \end{cases} \tag{3A}$$

$$\rightarrow p_2 = \frac{x_1}{x_1+x_2}(1-f_2-f_4) + \frac{x_3(x_1+x_5)}{(x_1+x_2)(x_3+x_4)}(f_2+f_4) \tag{4A}$$

Equating (2A) and (4A), one obtains:

$$x_1(x_3+x_4) = x_3(x_1+x_5) \tag{5A}$$

3. Short levers should be constrained so that they fit in the scale within the allowable bounds relative to the long levers position. Hence, using the cosine rule,

$$\begin{aligned} & x_3+x_4 \le \sqrt{a^2+b^2-2ab\cos\theta} \\ & where \\ & a = x_2 - x_5 \\ & b = \sqrt{x_7{}^2 + (.5x_{14}-y_1)^2} \\ & \theta = \pi - \mathbf{arctan}\left(\frac{.5x_{14}-y_1}{x_7}\right) - \mathbf{arccos}\left(\frac{x_{13}-2y_1-x_7}{x_1+x_2}\right) \end{aligned} \tag{6A}$$

$$\begin{aligned} & x_3+x_4 \ge \sqrt{d^2+e^2-2de\cos\phi} \\ & where \\ & d = x_2 - x_5 \\ & e = \sqrt{x_7{}^2 + y_5{}^2} \\ & \phi = \pi - \mathbf{arctan}\left(\frac{y_5}{x_7}\right) - \mathbf{arccos}\left(\frac{x_{13}-2y_1-x_7}{x_1+x_2}\right) \end{aligned} \tag{7A}$$

4. Furthermore, we should put some additional constraints to ensure that the angular terms achieve feasible values.

$$-1 \le \frac{x_{13}-2y_1-x_7}{x_1+x_2} \le 1 \tag{8A}$$

5. In the analog scale, the dial diameter should be restricted so that the dial does not reach to the support position on both levers. By defining the x as the distance from the dial center to the support position on the levers, we will have the following constraints:

$$x > x_{12}/2 \tag{9A}$$

5. 1. For long levers:

$$x = \sqrt{a^2+b^2-2ab\cos\theta}, \quad \begin{cases} a = x_{13} - 2y_1 - \dfrac{x_{12}}{2} - x_7, \; b = x_2 \\ \cos\theta = \dfrac{x_{13}-2y_1-x_7}{x_1+x_2} \end{cases} \tag{10A}$$

5.2. Short lever:

$$\begin{aligned} & x = \sqrt{a^2+b^2-2ab\cos\theta} \\ & a = x_4, b = \sqrt{a'^2+b'^2-2a'b'\cos\theta'} \end{aligned}$$

$$\begin{cases} a' = x_{13} - 2y_1 - \dfrac{x_{12}}{2} - x_7, \quad b' = x_2 - x_5 \\ \cos\theta' = \dfrac{x_{13}-2y_1-x_7}{x_1+x_2} \end{cases} \tag{11A}$$

$$\begin{aligned} & \theta = \theta_1 + \theta_2 \\ & \frac{a'}{\sin\theta_2} = \frac{b}{\sin\theta'} \rightarrow \theta_2 = Arc\sin\left(\frac{a'\sin\theta'}{b}\right) \end{aligned} \tag{12A}$$

$$\begin{aligned} & \theta'' = \theta_2 + \theta' - \frac{\pi}{2} \\ & l'' = b\cos\theta'', \quad h'' = b\sin\theta'' \\ & \delta = \sqrt{(x_3+x_4)^2 - (x_{13}-2y_1-\frac{x_{12}}{2}-h'')^2} \\ & l = l'' + \delta \\ & l' = \sqrt{l^2 + x_7^2} \\ & l'^2 = (x_3+x_4)^2 + (x_2-x_5)^2 - 2(x_3+x_4)(x_2-x_5)\cos\theta_1 \\ & \rightarrow \theta_1 = A\cos\left(\frac{(x_3+x_4)^2+(x_2-x_5)^2-l'^2}{2(x_3+x_4)(x_2-x_5)}\right) \end{aligned} \tag{13A}$$

6. In order to have a stable scale, the distance from the scale centerline to the support positions on both levers (S) is constrained to be more than the leg distance (y_7) from the center line.

$$S > y_7 \tag{14A}$$

6. 1. Long levers:

$$S = \frac{S' \times x_2}{x_1+x_2}, S' = \sqrt{(x_1+x_2)^2 - (x_{13}-2y_1-x_7)^2} \tag{15A}$$

6. 2. Short Levers:

6.2.1. Analog Scale:

$$S = l'' + \delta', \delta' = \frac{\delta \times x_4}{x_3+x_4} \tag{16A}$$

6.2.2 Digital Scale:

$$\begin{aligned} & S = l'' + \delta' \\ & \frac{a'}{\sin\theta_2} = \frac{b}{\sin\alpha} \rightarrow \theta_2 = Arc\sin\left(\frac{a'\sin\alpha}{b}\right) \\ & \theta'' = \theta_2 + \alpha - \frac{\pi}{2} \\ & l'' = b\cos\theta'', \quad h'' = b\sin\theta'' \\ & \delta = \sqrt{(x_3+x_4)^2 - (x_7+y_2+x_8+x_{10}-h'')^2} \\ & \delta' = \frac{\delta \times x_4}{x_3+x_4} \end{aligned} \tag{17A}$$

Appendix B: Pareto solutions

Table B1. Optimal product family for CI=2/12

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.87 8.67 2.74	2.86 8.39 2.70	1.74 9.96 2.41
Cover	11.42 11.91 0.75	10.58 13.23 0.56	12.00 10.00 0.68
Spring	169.93	179.99	100.80
Pivot	0.50 1.90	0.50 1.74	0.50 1.90
Short Lever	3.29 3.14	3.19 3.02	3.26 4.52
Rack & Pinion	6.80 0.25	6.59 0.25	6.80 0.25
Dial	8.9679	---	----
Weight Capacity	298.75	320.00	280.00
Aspect Ratio	0.961	0.8	1.2
Platform Area	135.97	140.00	120.00
Number Size	1.18		

Table B2. Optimal product family for CI=3/12

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.49 9.24 2.58	2.88 8.37 2.71	1.72 9.88 1.98
Cover	11.53 11.38 0.50	10.62 13.22 0.50	12.01 9.99 0.80
Spring	149.99	179.99	100.90
Pivot	0.50 1.90	0.50 1.74	0.50 1.90
Short Lever	3.36 3.48	3.18 2.99	3.78 4.36
Rack & Pinion	6.70 0.25	6.70 0.25	6.70 0.25
Dial	9.33	---	----
Weight Capacity	297.32	320.01	281.28
Aspect Ratio	1.01	0.80	1.20
Platform Area	131.24	140.42	120.01
Number Size	1.18		

Table B3. Optimal product family for CI=4/12

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.63 9.02 2.63	2.69 8.27 2.67	1.93 9.73 2.06
Cover	11.40 11.53 0.50	10.71 13.24 0.52	11.92 10.19 0.69
Spring	149.99	179.99	100.90
Pivot	0.51 1.89	0.50 1.79	0.51 1.89
Short Lever	3.25 3.23	3.25 3.23	3.76 4.01
Rack & Pinion	6.69 0.26	6.69 0.26	6.69 0.26
Dial	9.20	---	----
Weight Capacity	294.53	320.98	279.71
Aspect Ratio	0.99	0.81	1.17
Platform Area	131.38	141.86	121.46
Number Size	1.17		

Table B4. Optimal product family for CI=5/12

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.74 8.81 2.54	2.74 8.81 2.54	1.93 9.66 2.15
Cover	11.26 11.68 0.50	10.75 13.23 0.50	11.90 10.20 0.76
Spring	149.99	179.99	100.90
Pivot	0.50 1.89	0.50 1.80	0.50 1.89
Short Lever	3.28 3.04	3.28 3.04	3.68 4.10
Rack & Pinion	6.68 0.27	6.68 0.27	6.68 0.27
Dial	9.06	---	----
Weight Capacity	291.00	329.89	279.26
Aspect Ratio	0.96	0.81	1.17
Platform Area	131.47	142.18	121.41
Number Size	1.16		

Table B5. Optimal product family for CI=6/12

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.66 8.92 2.53	2.66 8.92 2.53	1.87 9.45 2.40
Cover	11.34 11.55 0.50	10.76 13.23 0.50	11.90 10.19 1.05
Spring	149.99	179.95	100.90
Pivot	0.50 1.88	0.50 1.88	0.50 1.88
Short Lever	3.32 3.15	3.32 3.15	3.35 4.30
Rack & Pinion	6.68 0.27	6.68 0.26	6.68 0.27
Dial	9.14	---	----
Weight Capacity	292.97	325.22	279.64
Aspect Ratio	0.98	0.81	1.17
Platform Area	130.87	142.33	121.23
Number Size	1.16		

Table B6. Optimal product family for CI=7/12

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.58 9.03 2.66	2.58 9.03 2.66	2.58 9.03 2.66
Cover	11.38 11.43 0.52	10.93 13.23 0.59	11.55 10.84 0.57
Spring	149.98	179.80	130.80
Pivot	0.50 1.89	0.50 1.89	0.50 1.89
Short Lever	3.24 3.32	3.24 3.32	3.25 3.44
Rack & Pinion	6.70 0.26	6.70 0.26	6.70 0.27
Dial	9.16	---	----
Weight Capacity	293.26	334.48	277.79
Aspect Ratio	1.00	0.83	1.07
Platform Area	130.09	144.51	125.27
Number Size	1.17		

Table B7. Optimal product family for CI=8/12

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.45 9.10 2.54	2.45 9.10 2.54	2.45 9.10 2.54
Cover	11.43 11.08 0.51	10.94 13.22 0.59	11.47 10.76 0.51
Spring	120.12	179.78	130.80
Pivot	0.50 1.90	0.50 1.90	0.50 1.90
Short Lever	3.31 3.43	3.31 3.43	3.31 3.43
Rack & Pinion	6.69 0.28	6.69 0.28	6.69 0.28
Dial	9.15	---	----
Weight Capacity	283.87	349.35	280.31
Aspect Ratio	1.03	0.83	1.07
Platform Area	126.61	144.72	123.43
Number Size		1.20	

Table B8. Optimal product family for CI=9/12

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.66 8.86 2.49	2.66 8.86 2.49	2.66 8.86 2.49
Cover	11.43 11.23 0.60	10.91 13.24 0.50	11.43 11.23 0.60
Spring	160.40	179.98	160.40
Pivot	0.50 1.88	0.50 1.88	0.50 1.88
Short Lever	3.42 3.20	3.42 3.20	3.42 3.20
Rack & Pinion	6.37 0.25	6.78 0.25	6.78 0.25
Dial	8.68	---	----
Weight Capacity	290.32	325.44	289.91
Aspect Ratio	1.02	0.82	1.02
Platform Area	128.31	144.45	130.34
Number Size	1.11		

Table B9. Optimal product family for CI=10/12

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.65 8.87 2.66	2.65 8.87 2.66	2.65 8.87 2.66
Cover	11.47 11.26 0.64	10.94 13.28 0.55	11.47 11.26 0.64
Spring	160.40	179.98	160.40
Pivot	0.50 1.87	0.50 1.87	0.50 1.87
Short Lever	3.27 3.27	3.27 3.27	3.27 3.27
Rack & Pinion	6.70 0.25	6.70 0.25	6.70 0.25
Dial	8.72	---	----
Weight Capacity	291.90	328.08	292.76
Aspect Ratio	1.02	0.82	1.02
Platform Area	129.17	145.27	129.20
Number Size	1.11		

Table B10. Optimal product family for CI=11/12

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	2.58 8.92 2.44	2.58 8.92 2.44	2.58 8.92 2.44
Cover	11.41 11.28 0.63	10.89 13.26 0.52	11.41 11.28 0.63
Spring	160.24	160.24	160.24
Pivot	0.50 1.85	0.50 1.85	0.50 1.85
Short Lever	3.44 3.25	3.44 3.25	3.44 3.25
Rack & Pinion	6.47 0.26	6.47 0.26	6.47 0.26
Dial	8.68	---	----
Weight Capacity	302.79	319.23	302.82
Aspect Ratio	1.01	0.82	1.01
Platform Area	128.68	144.47	128.68
Number Size	1.11		

Table B11. Optimal product family for CI=1

Component Name	Analog Scale	1st Digital Scale	2nd Digital Scale
Long Lever	1.84 9.18 2.20	1.84 9.18 2.20	1.84 9.18 2.20
Cover	11.48 11.37 0.50	11.48 11.37 0.50	11.48 11.37 0.50
Spring	119.98	119.98	119.98
Pivot	0.50 1.90	0.50 1.90	0.50 1.90
Short Lever	3.07 3.66	3.07 3.66	3.07 3.66
Rack & Pinion	6.80 0.25	6.80 0.25	6.80 0.25
Dial	9.28	---	----
Weight Capacity	296.39	296.39	296.39
Aspect Ratio	1.01	1.01	1.01
Platform Area	130.50	130.50	130.50
Number Size	1.17		