A Decomposed Gradient-Based Approach for Generalized Platform Selection and Variant Design in Product Family Optimization

Aida Khajavirad
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: aida@cmu.edu

Jeremy J. Michalek
Department of Mechanical Engineering,
and Department of Engineering and Public Policy,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: jmichalek@cmu.edu

A core challenge in product family optimization is to jointly determine (1) the optimal selection of components to be shared across product variants and (2) the optimal values for design variables that define those components. Each of these subtasks depends on the other; however, due to the combinatorial nature and high computational cost of the joint problem, prior methods have forgone optimality of the full problem by fixing the platform a priori, restricting the platform configuration to all-or-none component sharing, or optimizing the joint problem in multiple stages. In this paper, we address these restrictions by (1) introducing an extended metric to account for generalized commonality, (2) relaxing the metric to the continuous space to enable gradient-based optimization, and (3) proposing a decomposed single-stage method for optimizing the joint problem. The approach is demonstrated on a family of ten bathroom scales. Results indicate that generalized commonality dramatically improves the quality of optimal solutions, and the decomposed single-stage approach offers substantial improvement in scalability and tractability of the joint problem, providing a practical tool for optimizing families consisting of many variants. [DOI: 10.1115/1.2918906]

1 Introduction

An important goal in designing a successful product family is to exploit commonality across the family's product variants to reduce manufacturing cost without sacrificing the individual distinctiveness required to attract a variety of market segments. Hence, resolving the trade-off between commonality and the ability to achieve distinct performance targets has been the focus of many product family optimization studies. Simpson [1] reviewed 40 approaches to product family optimization. Most of these approaches have avoided the joint problem of simultaneously determining optimal (1) platform variable selection, (2) platform design, and (3) variant design, by fixing the platform a priori. However, optimal platform selection depends on design decisions; therefore, the optimal platform cannot typically be identified a priori without knowledge of the final design, and such approaches cannot, in general, offer optimality for the joint problem. Prior a posteriori approaches can be classified according to the number of stages involved: *Single-stage approaches* optimize both platform selection and design subproblems simultaneously, whereas *multi-stage approaches* select the platform in the first stage and fix the selection while optimizing the product family design in the second stage [1]. There is some trade-off between single- and two-stage approaches: Optimizing the platform and corresponding design variables in two separate stages may lead to suboptimal solutions, since the decisions are interdependent. However, single-stage approaches tend to have higher computational cost, which can make these algorithms impractical for large product families. Given these computational limitations, prior methods have restricted platform variable selection, insisting that a variable or component either be common across all variants or not at all. While this restriction improves problem tractability, it may lead to significant loss in performance, as we demonstrate later. In fact, many existing product families take advantage of *generalized commonality* [2], where components are shared among a subset of variants, so this restriction is often unrealistic. Therefore, an efficient and scalable single-stage approach is needed to avoid the suboptimal solutions of multistage methods and enable generalized commonality for product families of practical size.

Contributed by the Design Theory and Methodology Committee for publication in the Journal of Mechanical Design. Manuscript received July 18, 2007; final manuscript received October 19, 2007; published online May 19, 2008. Review conducted by Professor Janet K. Allen. Paper presented at the ASME 2007 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2007), Las Vegas, NV, September 4–7, 2007.

2 Background

2.1 Commonality Metrics. Several indices and metrics have been developed to measure the degree of commonality in a product family. These studies can be classified into two main categories according to their development purpose and application: (1) commonality *indices* for evaluation of existing product lines and (2) commonality *metrics* for use in product family optimization.

The first group of studies proposes commonality indices for measuring commonality in existing product lines to compare product families or assess improvement of a redesign. Some of these indices consider only the number of common parts within the family [3–5] while others consider specific information about the product line, such as production volume and component costs [6–9]. All aforementioned indices represent a proper commonality measurement with respect to the related criteria. Therefore, selection of the appropriate index involves consideration of the company focus and standpoint when designing the product family.

In the second category, commonality metrics are defined within the context of optimization for resolving the trade-off between the commonality and achievement of distinct performance targets. Due to the high computational cost of the joint problem, several approaches have restricted the commonality definition to *all-or-none* component sharing [10–15]: That is, a component can either be common within the entire family or be distinct among all prod-

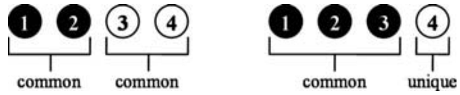

Fig. 1 The double-counting property of Fellini's metric; (left) case 1: $\eta=2$; (right) case 2: $\eta=3$

ucts but cannot be shared among only a subset of the variants. This simplification imposes an unnecessary restriction for many cases, leading to suboptimal solutions. Fellini et al. [16] addressed the above restriction by defining the commonality metric as the summation of all possible pairwise comparisons within the product family, assigning a binary variable to each pair that is equal to 1 if the corresponding components are shared and 0 otherwise. However, while this commonality definition allows the possibility of having multiple "subplatforms," we argue that it is limited for practical application due to its *double-counting* property. This shortcoming can be best illustrated with an example: Consider the two alternative platform configurations in Fig. 1 for a single component (module) within a family of four variants (other components are not shown). Here, 1–4 represent the component of interest for each variant, and the colors indicate the component sharing. In both cases, two distinct component designs must be produced, and two sets of tooling must be purchased; therefore, both alternatives may be considered equivalent with respect to tooling cost benefits of commonality.[1] However, Fellini's metric gives preference to the second case,[2] describing the first case as commonality level $\eta=2$ and the second case as commonality level $\eta=3$. Hence, this pairwise comparison-based metric prefers configurations that group more components into the same subplatform and as a result, produces a convergence bias toward platform architectures with all-or-none component sharing.

In brief, while the commonality indices introduced in the first group appear more realistic in measuring the degree of commonality within an existing family, they have not been applied in the optimization context, since they require a given product family structure, which is unavailable during the optimization process. Commonality metrics in the second group restrict the commonality to all-or-none component sharing or suffer from double counting; therefore, they are incapable of measuring the benefits of commonality for practical cases properly. Hence, an approach is needed to adapt indices from the first group for use in optimization.

2.2 Prior Single-Stage Approaches for Solving the Joint Problem. Most prior single-stage approaches use genetic algorithms (GAs) for solving the joint problem [10,11,15,17–20]. However, applying stochastic methods such as GAs involves significant computational cost and a lack of scalability for dealing with large problems. Khajavirad et al. [20] proposed a decomposed GA that improves scalability; however, while GAs offer global search, their stochastic nature limits the ability to ensure local or global optimality and requires significant time for problem-specific parameter tuning. Therefore, for cases where variant design can be analytically formulated, gradient-based methods are preferred in that they guarantee at least local optimality (and global optimality for convex problems) and are computationally efficient. Khire et al. [12] applied the selection integrated optimization method for solving the joint problem with gradient-based methods by formulating the combinatorial problem as a series of continuous relaxations. While the proposed method proved to be robust for optimizing the joint problem in a single stage, it hinges on the all-or-none commonality restriction. A single-stage gradient-based approach is needed that can solve the joint problem using generalized commonality with a reasonable computational cost.

2.3 Decomposition. One approach for exploiting the special structure of certain classes of large-scale optimization problems is to decompose the original problem into a number of smaller subsystems that are separately optimized and coordinated to arrive at the overall system optimum. In particular, *analytical target cascading* (ATC) was specifically developed for solving hierarchies of interacting systems and subsystems, and convergence proofs are available [21–25]. Hence, it has been widely used for optimizing engineering design problems with hierarchical structures [26–30]. Kokolaras et al. [28] noticed the hierarchical structure of the product family optimization problem and applied target cascading to the design of a product family with a fixed (a priori) platform configuration. Michalek et al. [30] also applied ATC to product line design using market data to predict demand and revenue and manufacturing models to predict cost; however, the approach did not address commonality among products in the line. Therefore, a decomposition scheme that exploits the special structure of the joint problem is crucial to make the single-stage approach more practical and scalable.

2.4 Proposed Approach. In Sec. 3, we propose a generalized commonality metric and develop a single-stage, relaxation- and decomposition-based optimization approach for solving the joint selection and design problem using gradient-based methods. In Sec. 4, we demonstrate the approach with a case study involving a family of ten bathroom scales. Results and conclusions are presented in Secs. 5 and 6, respectively.

3 Proposed Methodology

The basic formulation for optimizing a single product can be extended for optimizing a family of products by considering commonality decisions as consistency constraints. In order to find the optimal platform configuration and corresponding set of products simultaneously, the commonality metric and decision variables are also added to the original formulation. Hence, for a product family with n products, each with m components, the joint problem can be formulated as:

$$\begin{aligned}
&\text{Maximize} \quad \{\Gamma(\eta), \mathbf{f}_i(\mathbf{x}_i)\ \forall\ i\}\\
&\text{with respect to} \quad \eta_{ijk}, \mathbf{x}_{ik}\\
&\text{subject to} \quad \mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{0}, \quad \mathbf{h}_i(\mathbf{x}_i) = \mathbf{0}\\
&\eta_{ijk} = \begin{cases} 1 & \text{if } \mathbf{x}_{ik} = \mathbf{x}_{jk} \\ 0 & \text{otherwise} \end{cases}\\
&\forall\, i,j = \{1,2,\ldots,n\}, \quad k = \{1,2,\ldots,m\}\\
&\text{where} \quad \eta = \{\eta_{ijk}\ \forall\ i,j,k\}, \quad \mathbf{x}_i = \{\mathbf{x}_{ik}\ \forall\ k\}
\end{aligned} \tag{1}$$

where Γ is the commonality objective, $\mathbf{x}_{ik}$ is the set of variables for product i that define component (module) k, and $\mathbf{f}_i$, $\mathbf{g}_i$, and $\mathbf{h}_i$ represent the vectors of performance objective functions, inequality and equality constraints for the ith product, respectively. The last relation in Eq. (1), which we call the *commonality consistency constraint*, assigns a binary value η_{ijk} to each component k that is equal to 1 if the vector of design variables for that component is shared between the corresponding product pair ($\mathbf{x}_{ik}$, $\mathbf{x}_{jk}$) and 0 otherwise.

3.1 Commonality Index. In Eq. (1), $\Gamma(\eta)$ measures commonality within the entire family. Defining the proper form for Γ depends on the company's perspective when designing a product family. In this study, we consider the commonality benefit due to

[1] If more information is known about the production volume of each variant and the life of the tooling, a more accurate prediction can be made; however, commonality metrics are generally applied at a higher level of abstraction so that they do not require excessive data to compute.

[2] If anything, the first alternative would probably be preferred over the second because the sharing appears to be more balanced (again, this depends on production volume).

tooling cost savings and adopt the commonality index (CI) introduced by Martin and Ishii [4], which is a measure of the percentage of unique parts,

$$\mathrm{CI} = 1 - \frac{u - \max_i(m_i)}{\sum_{i=1}^{n} m_i - \max_i(m_i)} \tag{2}$$

where u is the total number of distinct component designs in the product family and m_i is the number of components in the ith product. CI ranges from 0 to 1, and a higher value indicates fewer unique parts.[3] In computing CI from Eq. (2), it is assumed that the product family structure is given. Hence, in order to apply the metric within an optimization context (i.e., Eq. (1)), it should be reformulated as a function of the binary commonality variables. Let us reconsider the commonality representation in Eq. (1). For the kth component in the product family, we define the commonality matrix $\mathbf{\Gamma}_k$ as follows:

$$\mathbf{\Gamma}_k \equiv \begin{bmatrix} 1 & \eta_{12k} & \cdots & \eta_{1nk} \\ \eta_{21k} & 1 & & \eta_{2nk} \\ \vdots & & \ddots & \vdots \\ \eta_{n1k} & \eta_{n2k} & \cdots & 1 \end{bmatrix} \tag{3}$$

In Eq. (3), it is assumed that all variants include an instance of the kth component; however, for the general case that some variants may not include all components, the corresponding binary commonality variables in Eq. (3) are set to zero. Imposing the transitivity constraints on commonality variables to ensure a consistent matrix (e.g., if $\eta_{12}=1$, $\eta_{23}=1 \rightarrow \eta_{13}=1$), the set of platforms for component i is well defined, and Eq. (3) can be rearranged to the following block diagonal format:

$$\mathbf{\Gamma}_k = \begin{bmatrix} [\mathbf{1}]_{n_{k1}\times n_{k1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & [\mathbf{1}]_{n_{kr}\times n_{kr}} & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & [\mathbf{1}]_{n_{ks_k}\times n_{k_{ks_k}}} \end{bmatrix} \tag{4}$$

where $[\mathbf{1}]_{n_{kr}\times n_{kr}}$ represents an $n_{kr}\times n_{kr}$ matrix with all elements equal to 1, and $r=\{1,2,\ldots,s_k\}$ is the index of subplatforms for component k. Furthermore, the number of these submatrices s_k, also called "blocks," is equal to the number of distinct platforms for producing the kth component:

$$\sum_{k=1}^{m}\sum_{r=1}^{s_k} n_{kr} = nm \tag{5}$$

Eigenvalues of a block diagonal matrix are simply those of its blocks; hence, Eq. (4) has s_k *nonzero* eigenvalues λ_{kr}, each equal to the number of variants in the rth subpaltform of the kth component:

$$\lambda_{kr} = n_{kr}, \quad r = 1, \ldots, s_k \tag{6}$$

The total number of unique components u in the family can be found from the following:

$$u = nm - \sum_{k=1}^{m}\sum_{r=1}^{s_k} (n_{kr} - 1) \tag{7}$$

Substituting Eqs. (5)–(7) into Eq. (2), CI can be reformulated as follows:

$$\mathrm{CI} = \frac{\sum_{k=1}^{m}\sum_{r=1}^{s_k}(\lambda_{kr} - 1)}{m(n-1)} \tag{8}$$

[3]If data are available on relative cost savings for commonality of each component, appropriate weights can be included as well.

Using the above formula, CI is defined as a function of the binary commonality variables, i.e., the form that can be computed during the optimization procedure. However, in practice, it is not convenient to solve Eq. (1) in the mixed-integer format. Hence, in the following sections, Eq. (8) will be revised and extended to the continuous space, and alternative functional forms for relaxing binary commonality variables will be investigated.

3.2 Extended Commonality Index. In deriving CI using the discrete format, use of only the nonzero λ_{kr} terms omits negative terms from Eq. (8). However, by relaxing CI to the continuous space, this condition is no longer valid, and Eq. (8) should be modified as follows:

$$\mathrm{ECI} = \frac{\sum_{k=1}^{m}\sum_{r=1}^{s_k}(\max(\tilde{\lambda}_{kr}, 1) - 1)}{m(n-1)} \tag{9}$$

where ECI is the extended commonality index to continuous space and $\tilde{\lambda}_{kr}$ is the rth eigenvalue of the relaxed commonality matrix for the kth component. Equation (9) introduces a discontinuity in the derivative of ECI. While it is possible to eliminate the discontinuity through introduction of slack variables and binary variables, our empirical examples suggest that such reformulation is unnecessary, since gradient-based algorithms perform well with the form of Eq. (9).

In order to investigate the effect of relaxation on ECI, two basic cases are considered in the relaxed space: (1) a distinct component becomes common with an existing platform and (2) a component deviates from the platform to which it belongs and become common to another platform. The case of a platform component becoming distinct is addressed by the first case. Consider a platform with n' components within the product family and an arbitrary component that is initially distinct from this platform. The submatrix associated with this augmented platform in discrete format is as follows:

$$\mathbf{\Gamma}_{kr} = \begin{bmatrix} 1 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & 0 \\ 1 & \cdots & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{(n'+1)\times(n'+1)} \tag{10}$$

Relaxing the distinct variable to the continuous space, it becomes common with the platform components by the value of γ $(0 \leqslant \gamma \leqslant 1)$. Therefore, the new sub-matrix will become

$$\mathbf{\Gamma}'_{kr} = \begin{bmatrix} 1 & \cdots & 1 & \gamma \\ \vdots & \ddots & \vdots & \gamma \\ 1 & \cdots & 1 & \gamma \\ \gamma & \gamma & \gamma & 1 \end{bmatrix}_{(n'+1)\times(n'+1)} \tag{11}$$

Equation (11) has two nonzero eigenvalues, which can be computed from the following relations:

$$\lambda'_1 = \left((n'+1) + \sqrt{(n'+1)^2 - 4n'(1-\gamma^2)}\right)/2$$

$$\lambda'_2 = \left((n'+1) - \sqrt{(n'+1)^2 - 4n'(1-\gamma^2)}\right)/2 \tag{12}$$

Substituting the above eigenvalues into Eq. (9), ECI can be found from the following:

$$\mathrm{ECI} = \frac{0.5\left((n'+1) + \sqrt{(n'+1)^2 - 4n'(1-\gamma^2)}\right) - 1}{n'} \tag{13}$$

Similarly, for the second case originally having two platforms with $n'+1$ and n' components, respectively, the result is

$$\mathrm{ECI} = \frac{0.5\left(3(n'-1) + \sqrt{(n'+1)^2 - 8n'\gamma(1-\gamma)}\right)}{2n'} \tag{14}$$

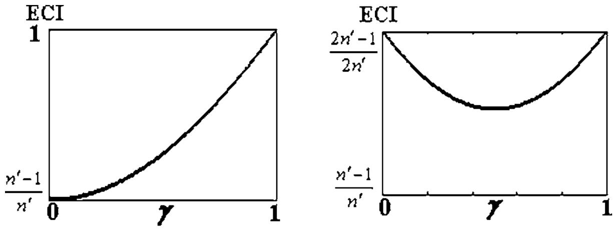

Fig. 2 Commonality level change for two basic cases

Figure 2 shows ECI for the above cases (Eqs. (13) and (14)) with respect to γ. The extended metric has desirable properties: It attains correct values at integers and provides a smooth continuous transition for intermediate values of the relaxation.

Any change in the product family architecture during the optimization process can be regarded as a combination of these basic cases. It should be noted that, since the system is nonlinear (as can be seen from the eigenvalue equations), the superposition principle cannot be applied for a general case. However, the analytical results for these simplified cases along with the numerical justifications for general product families indicate the appropriateness of the proposed metric for relaxation.

3.3 Relaxation of the Commonality Consistency Constraint. After defining a commonality metric that remains valid in the continuous space, the binary commonality consistency constraint should be relaxed using a continuous, differentiable function that ranges from 0 to 1 with correct values at the end points. Fellini et al. [16] proposed the following approximating function:

$$\eta'_{ijk} = \frac{1}{1 + (\Delta \mathbf{x}_{ijk}/\alpha)^2}, \quad \Delta \mathbf{x}_{ijk} = \|\mathbf{x}_{ik} - \mathbf{x}_{jk}\| \tag{15}$$

where α is a small value greater than zero that controls the degree to which the curve approximates the discontinuous step function: As α decreases, the optimal solution of the continuous problem tends toward that of the discrete formulation. However, as α approaches zero, Eq. (15) becomes ill conditioned, which leads to numerical difficulties. Therefore, the approach is to approximate the discrete formulation by a sequence of continuous optimization problems in which α is iteratively decreased until variables that are designated as common fall within an acceptable deviation tolerance. To support convergence, it is desirable that η should approach its final value for α small but sufficiently greater than zero to prevent ill conditioning, so it is desirable that $\partial\eta/\partial\alpha \approx 0$ for small α. In addition, since Eq. (9) depends on the approximating function, it is desirable to select the function so that ECI is continuous and differentiable. We propose two alternatives to Eq. (15), which are then compared with respect to the aforementioned criteria: The logistic curve has been used frequently in probabilistic models for pairwise comparison. It can be applied as a relaxation function with slight changes to the standard form, which we call the *half logistic curve* for distinction:

$$\eta'_{ijk} = \frac{2e^{-\Delta \mathbf{x}_{ijk}/\alpha}}{1 + e^{-\Delta \mathbf{x}_{ijk}/\alpha}}, \quad \Delta \mathbf{x}_{ijk} = \|\mathbf{x}_{ik} - \mathbf{x}_{jk}\| \tag{16}$$

Another candidate is the *Hubbert curve*, which is the derivative of the logistic function:

$$\eta'_{ijk} = \frac{4e^{-\Delta \mathbf{x}_{ijk}/\alpha}}{(1 + e^{-\Delta \mathbf{x}_{ijk}/\alpha})^2}, \quad \Delta \mathbf{x}_{ijk} = \|\mathbf{x}_{ik} - \mathbf{x}_{jk}\| \tag{17}$$

The first derivative of Eqs. (15)–(17) ($\partial\eta'/\partial\alpha$) are plotted in Fig. 3 as a function of α for a fixed deviation ($\Delta\mathbf{x}=0.01$).

As can be found from Fig. 3, for both the half logistic and Hubbert curves, the first derivative approaches zero for $\alpha \leqslant 0.001$. However, for Fellini's proposed function, the curve does not approach a constant value even for $\alpha \leqslant 0.0005$, which leads to convergence problems during the optimization process. Next, in order to investigate the effect of the approximating function on ECI characteristics, the three candidate functions are plugged into Eq. (9). Here, for simplicity and better visualization, three distinct components are considered. We are interested to sketch the commonality change for a sequence of decreasing α values as a function of one component x_3, given fixed values for the other components x_1 and x_2. Over the range of possible values for x_3, the component begins distinct, then becomes common with the first component x_1, deviates and becomes common with the second component x_2, and finally deviates again to become distinct (Fig. 4); as can be seen from this figure, in the case of the half logistic curve, ECI is not differentiable at the two extreme points, which may cause numerical difficulties for optimization. However, for both Hubbert and Fellini's curves, ECI has a continuous derivative and is concave in the neighborhood of the local optima. Hence, according to the two aforementioned criteria, the Hubbert curve shows the best characteristics, since it converges to the discrete solution as α decreases and results in a differentiable ECI with respect to $\mathbf{x}$.

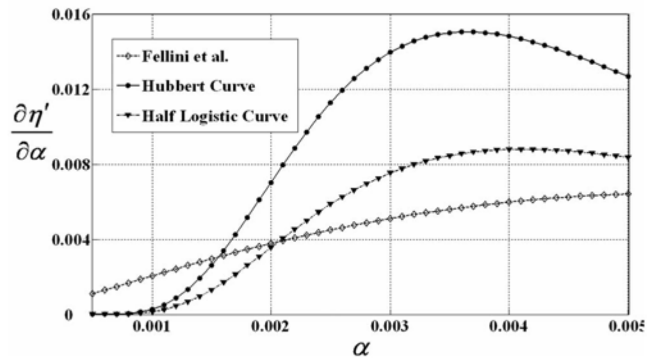

Fig. 3 First derivative of the approximating functions with respect to α

3.4 Proposed Approach: All-in-One Formulation. Using ECI (Eq. (9)) as the commonality metric and the Hubbert curve (Eq. (17)) as the relaxation function, the MINLP formulation in Eq. (1) can be replaced by a sequence of NLP optimizations as follows:

$$\text{Maximize} \quad \mathbf{f}_i(\mathbf{x}_i), \quad i = 1, \ldots, n$$

$$\text{Maximize} \quad \text{ECI}(\mathbf{\Gamma}'_k, \alpha_s), \quad k = 1, \ldots, m$$

$$\mathbf{\Gamma}'_k = \begin{bmatrix} 1 & \eta'_{12k} & \cdots & \eta'_{1nk} \\ \eta'_{21k} & 1 & & \eta'_{2nk} \\ \vdots & & \ddots & \vdots \\ \eta'_{n1k} & \eta'_{n2k} & \cdots & 1 \end{bmatrix}, \quad \eta'_{ijk} = \frac{4\exp\left\| \dfrac{\mathbf{x}_{ik} - \mathbf{x}_{jk}}{\alpha_s} \right\|}{\left(1 + \exp\left\| \dfrac{\mathbf{x}_{ik} - \mathbf{x}_{jk}}{\alpha_s} \right\|\right)^2}$$

$$\text{with respect to } \mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$$

$$\text{subject to } \mathbf{g}_i(\mathbf{x}_i) \le \mathbf{0}, \quad \mathbf{h}_i(\mathbf{x}_i) = \mathbf{0}$$

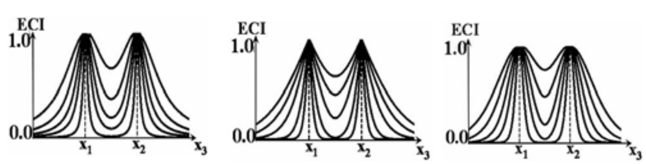

Fig. 4 ECI using three approximating functions; (left) Fellini's curve; (middle) Half Logistic; (right) Hubbert curve

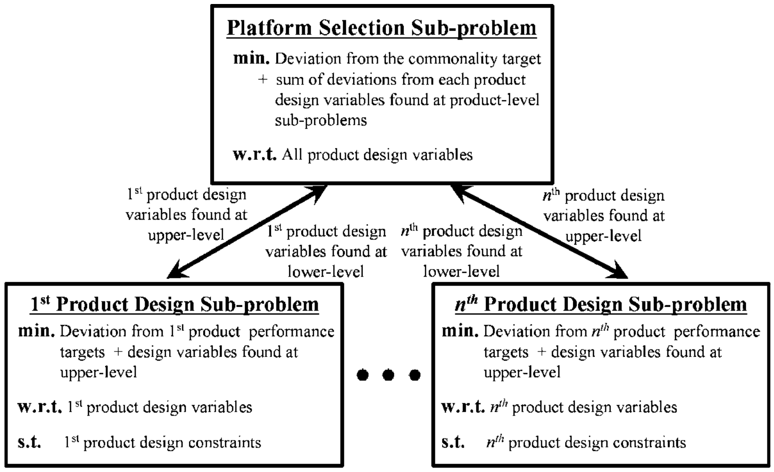

Fig. 5 ATC framework for optimizing the joint product family problem

$$\text{where } \alpha_{s+1} = c\alpha_s, \quad 0 < c < 1 \tag{18}$$

Hence, by defining α_0 and c, Eq. (18) is iteratively optimized until the difference between the common variables fall within an acceptable tolerance. In Eq. (18), we used a linear scheme for decreasing α. However, in general, different methods, such as an exponential reduction scheme, can be applied depending on the form of the approximating function. Moreover, selection of appropriate values for c is problem specific and should be properly tuned for each case. Equation (18) is a multiobjective optimization problem with $1+\sum_{i=1}^{n} p_i$, objective functions, where p_i is the number of objective functions for the ith product. In practice, we are interested to sketch the Pareto frontier for capturing the trade-off between increasing commonality and loosing variant performance. Hence, all performance objectives can be grouped into one objective defined as the (normalized and possibly weighted) sum of the deviation of each achieved performance level $\mathbf{f}(\mathbf{x}^i)$ from its corresponding performance target $\mathbf{T}_i$. Moreover, using the discrete definition for commonality variables, CI attains the following values:

$$\mathrm{CI} = \frac{r}{m(n-1)}, \quad r \in \{0, \ldots, m(n-1)\} \tag{19}$$

Therefore, the multiobjective optimization problem can be converted to a series of $1+m(n-1)$ single objective optimizations, each finding the optimal platform and individual product designs for a fixed commonality level l^r. Applying the above modifications, Eq. (18) can be reformulated as follows:

$$\text{Minimize} \quad (l^r - \mathrm{ECI}(\boldsymbol{\Gamma}'_k, \alpha_s))^2 + \sum_{i=1}^{n} w_i \|\mathbf{T}_i - \mathbf{f}_i(\mathbf{x}_i)\|_2^2, \quad k = 1, \ldots, m$$

$$l^r = \frac{r}{m(n-1)}, \quad r \in \{0, \ldots, m(n-1)\}$$

$$\text{subject to } \mathbf{g}_i(\mathbf{x}_i) \le \mathbf{0}, \quad \mathbf{h}_i(\mathbf{x}_i) = \mathbf{0}$$

$$\alpha_{s+1} = c\alpha_s, \quad 0 < c < 1 \tag{20}$$

where w_i is weighting coefficient showing the relative importance of performance objectives; $\|\cdot\|_2^2$ denotes the square of the l_2 norm; and $\boldsymbol{\Gamma}'_k$ and η'_{ijk} are computed as defined in Eq. (18).

3.5 Proposed Approach: Decomposition Scheme. Increasing the problem size makes the all-in-one formulation (Eq. (20)) impractical for solving the joint problem directly, and efficiencies may be gained through decomposition. According to the ATC framework, the original all-in-one problem with a hierarchical structure is decomposed into a top level system and a hierarchy of subsystems. For every output of each subsystem that affects the parent system, two copies are created: a target copy x managed by the system and a response copy y managed by the subsystem. An ATC consistency constraint function $\pi(x\text{-}y)$ is added to ensure that $x=y$ at the solution, and π is relaxed using penalty functions or Lagrangian relaxation to support decomposition and coordinate a consistent and optimal solution (see Ref. [25] for a comprehensive review).

As can be seen from Eq. (20), the only nonseparable part in the all-in-one formulation is the commonality deviation portion of the objective function. Hence, using ATC the joint product family problem can be decomposed to a two-level optimization problem: The system level optimization problem finds the optimal platform configuration while each subsystem deals only with optimizing a single product in the family (Fig. 5).

The resulting system level problem is an unconstrained NLP problem that finds the optimal platform and distinct design variables $\mathbf{x}$ for a given commonality level l_r and with minimum deviation from the responses $\mathbf{y}$ passed up from the product level subsystems:

$$\text{Minimize} \quad (l^r - \mathrm{ECI}(\boldsymbol{\Gamma}'_k, \alpha_s))^2 + \sum_{i=1}^{n} \pi(\mathbf{x}_i - \mathbf{y}_i), \quad k = 1, \ldots, m$$

$$l^r = \frac{r}{m(n-1)}, \quad r \in \{0, \ldots, m(n-1)\}$$

$$\text{with respect to } \mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$$

$$\alpha_{s+1} = c\alpha_s, \quad 0 < c < 1 \tag{21}$$

where $\boldsymbol{\Gamma}'_k$ and η'_{ijk} are computed as defined in Eq. (18). The optimization problem for the ith subsystem, which optimizes the ith product, has the following formulation:

$$\text{Minimize} \quad w_i \|\mathbf{T}_i - \mathbf{f}_i(\mathbf{y}_i)\|_2^2 + \pi(\mathbf{y}_i - \mathbf{x}_i)$$

$$\text{with respect to } \mathbf{y}_i$$

$$\text{subject to } \mathbf{g}_i(\mathbf{y}_i) \leqslant \mathbf{0}, \quad \mathbf{h}_i(\mathbf{y}_i) = \mathbf{0} \tag{22}$$

where $\mathbf{y}_i$ is the vector of local variables representing the design variables of the ith product, and $\mathbf{x}_i$ is the target vector cascaded

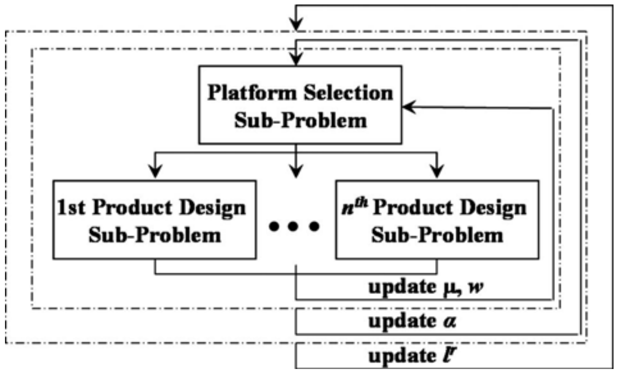

Fig. 6 Decomposition algorithm for optimizing the joint product family problem

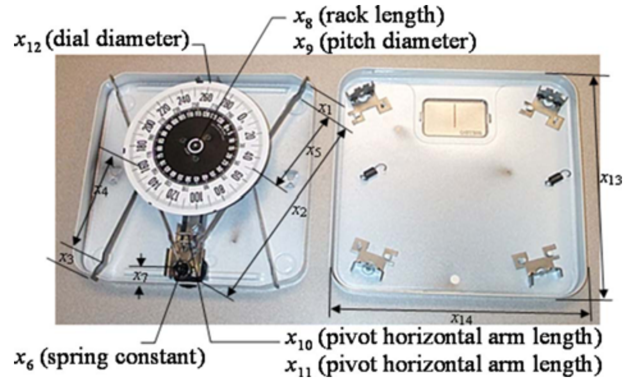

Fig. 7 Design variables shown on the disassembled analog scale [29]

down from the system level problem.[4]

A variety of approaches have been used to decompose and coordinate consistency among subsystems [21–25]. We adopt the augmented Lagrangian alternating direction (ALAD) method of multipliers [24]: According to this method, the ATC consistency constraint relaxation function $\pi(\mathbf{y}_i-\mathbf{x}_i)$ is the augmented Lagrangian function $\boldsymbol{\mu}^T(\mathbf{y}_i-\mathbf{x}_i)+\mathbf{w}^T(\mathbf{y}_i-\mathbf{x}_i)^2$, where the values of $\boldsymbol{\mu}$ are iteratively updated using the method of multipliers. In the ALAD approach, each subproblem is solved only once before updating $\boldsymbol{\mu}$ via the method of multipliers, and the penalty weight $\mathbf{w}$ may be held constant or only be updated when no improvement in the objective function is observed.

The optimization scheme for solving the decomposed problem is sketched in Fig. 6: In the inner loop, for a fixed α and commonality value, the ATC formulation is solved using the ALAD method. In the middle loop, α iteratively decreases until the relative difference among the shared components falls below user specified tolerances. Finally, after finding the optimal platform configuration and individual products for a constant commonality value, l^r is updated, and this procedure continues until the optimum product families for the entire range of commonality levels are found. Thus, the inner loop handles coordination of subsystems in the decomposition, the middle loop drives the commonality relaxation toward an integer solution, and the outer loop generates points along the Pareto frontier.

4 Case Study: Bathroom Scale Design

We now apply the proposed approach to the design of a family of standard dial-read out and digital bathroom scales from the literature. Design of a family of scales is a well-suited example for illustrating the trade-off between commonality and achievement of distinct performance targets[5] since individual products with distinct characteristics (e.g., digital and analog) operate according to nearly identical principles, leading to significant market differentiation with high potential levels of engineering commonality. The engineering optimization model is taken from Ref. [29]. Product design variables are depicted in Fig. 7 on the analog scale, which is the same for the digital scale excluding the dial diameter (the dial is replaced with an encoder wheel and photointerrupter).

[4]The proposed decomposition improves scalability by (1) separating platform selection from variant design and (2) optimizing each product separately. However, the size of the remaining platform selection subproblem constrains the degree of scalability for large problems.

[5]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 trade-off [30,31]; however, we do not pursue this here.

Table 1 Scale components and their design variables[a]

	Component name	Associated variables
1.	Long Lever	$\{x_1,x_2,x_5\}$
2.	Short Lever	$\{x_3,x_4\}$
3.	Spring	$\{x_6\}$
4.	Rack and Pinion	$\{x_8,x_9\}$
5.	Pivot	$\{x_{10},x_{11}\}$
6.	Cover	$\{x_7,x_{13},x_{14}\}$

[a]Dial commonality among the analog scales is ignored since the dial production cost is negligible compared to other components.

For module-based product family optimization, design variables are grouped according to the component (module) to which they belong (Table 1) and, the commonality is measured based on component sharing; i.e., two products have a common part if *all* of the design variables corresponding to that part have the same value for both products.

Performance objectives are weight capacity, platform area, aspect ratio, and number size (analog scale). To capture several design issues ignored in the prior study [29], the following constraints were added to the optimization formulation:[6]

1 Maximum displacement of the spring must remain below the allowable value restricted by the scale thickness.
2 The scale should be designed so that it measures the right weight regardless of the user's leg positions.
3 Short levers should be constrained so that they fit in the scale within the allowable bounds relative to the long lever positions.
4 In the analog scale, the dial diameter should be restricted so that the dial does not interfere with the support position on either lever.
5 The distance from the scale centerline to the support positions on both levers is constrained to be more than the leg distance from the center line for stability.

5 Results

In order to test the effects of generalized commonality and the scalability of the decomposed single-stage method, a family of ten bathroom scales including five analog and five digital scales was optimized. Individual product performance targets were selected from the product attribute levels suggested in Ref. [29] to define a set of distinct, feasible products.

[6]The complete mathematical formulation is available in Ref. [29] or from the authors. Details were not included here due to space limitations.

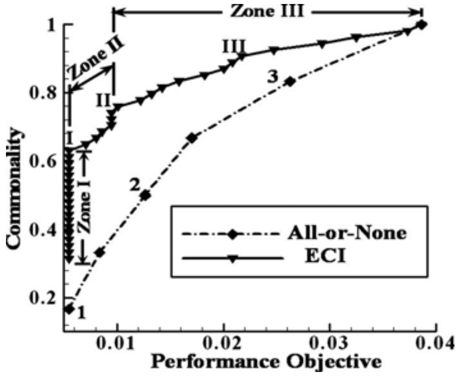

Fig. 8 Pareto curves for family of ten bathroom scales

First, individual variants were separately optimized to find optimal designs under the condition of no enforced commonality. Although no commonality was enforced in this case, separately optimized products result in ECI$=17/54$ for this problem. These designs were applied as initial seeds for product family optimization. Next, the ATC framework (Fig. 5) was applied for optimizing a family of ten bathroom scales with $a_0=0.01$ decreased in ten steps by a factor of $c=0.75$. The overall performance objective is formulated as the average of the l_2 norm of the normalized deviation vector of individual attributes from their corresponding performance targets over the entire family. For comparison, ECI (Eq. (10)) and the restricted all-or-none version were separately applied. The all-or-none commonality metric was computed by replacing the normalized pairwise difference $\Delta\mathbf{x}$ in Eqs. (15)–(17) with the variance of the kth component across the variants. The resulting Pareto frontiers are plotted in Fig. 8. While both optimal fronts depict the inherent trade-off between commonality and the ability of the variants to achieve distinct performance targets, the generalized curve dramatically dominates the all-or-none curve. That is, for a given level of performance loss, a much higher commonality value can be achieved using ECI. For example, Figure 8 shows that for achieving the minimum performance loss using the restricted metric, the commonality level can only be increased up to 17% (i.e., shared pivot among all variants); however, by allowing component sharing within subsets of products, the commonality level can be increased to 62% for the same performance loss, leading to significant cost savings.

Figure 8 divides the Pareto frontier for the generalized case into three segments: In Zone I, it is possible to increase the commonality up to 62% without any performance loss, which reveals (1) the importance of solving the joint product family problem and (2) the importance of including generalized commonality. In Zone II, for $0.62 < \text{ECI} < 0.78$, the rate of performance loss is relatively slow; that is, product attributes deviate only nominally from the assigned targets. However, in Zone III, the performance loss rapidly increases, and individual products cannot achieve their target characteristics. Therefore, by plotting the product family Pareto front, the designer is provided with a helpful perspective on how to decide the desired commonality level and its corresponding platform configuration to reduce manufacturing cost without excessive sacrifice of distinctiveness.

Table 2 describes the platform configuration for three points on the Pareto frontier and their nearest comparison points on the restricted curve. The numbers listed indicate the number of variants that share each component design (with 1's omitted). For example, the notation {6,2} indicates that one design is shared among six variants, a second design is shared among two variants, and the remaining two variants have distinct designs. As can be seen from this table, the platform configuration (and design variable values, not listed here) are such that the optimal family with commonality level r does not necessarily involve the platform variables optimal for commonality level $r-1$. This result highlights the importance of solving the joint problem instead of selecting the platform architecture a priori

Table 2 Platform configurations for points on the Pareto frontier (Fig. 8)

	All-or-none			ECI		
Module	1	2	3	I	II	III
Long lever	—	—	10	4	6,2	8,2
Short lever	—	10	10	4,4,2	7,2	10
Spring	—	—	10	5,3	6,4	10
Rack and pinion	—	10	10	6,3	7	9
Pivot	10	10	10	10	10	10
Cover	—	—	—	2,2	2,2,2	3,3,2,2

Table 3 lists the total computational time required to solve for each of the three points shown in Fig. 8 using alternative schemes (the restricted commonality case uses Points 1, 2, and 3 instead). Overall, the generalized commonality formulation produces improved solutions; however, higher complexity of the commonality objective and search space increases computational requirements, motivating the need for decomposition. As can be seen from the table, for the bathroom scale example, decomposing the joint problem using ATC decreases computational requirements by 50–60%. In addition, further speedup can be achieved by optimizing the individual subsystems in parallel, which provides ~70% improvement over the all-in-one case. In the current case study, each variant design formulation is relatively simple, and computational time of the ATC framework is dominated by the upper level subproblem. In cases where the variant subproblems have greater complexity, higher speedups due to the decomposition and parallelization would be expected.

6 Summary and Conclusions

In this study, we first proposed a method for computing the CI introduced by Martin and Ishii for use in optimization and argued for its improved properties over prior metrics applied for optimization. The discrete definition of the CI was then extended to the continuous space, and the properties of the index were examined. Results show that the proposed metric remains valid in the continuous space, enabling relaxation of the MINLP formulation into a continuous domain, which enables use of gradient-based approaches. Considering the critical effect of the approximating consistency function on ECI performance, two important criteria for selecting the proper function were described, and three alternatives were compared. The Hubbert curve proved to be the only alternative possessing the desired characteristics. Therefore, by applying the ECI as the commonality objective along with the Hubbert curve as the approximating function, a novel single-stage gradient-based approach for optimizing the joint platform selection and design problem was introduced. In order to address the scalability of the proposed method, the all-in-one formulation was

Table 3 Effect of generalization, decomposition, and parallelization on the computational time (seconds)

	All-or-none	ECI		
			ATC	
Optimization scheme	ATC (serial)	All-in-one	Serial	Parallel
Point I (1)	(115)	1294	523	371
Point II (2)	(232)	1392	655	454
Point III (3)	(453)	1500	702	472

decomposed using ATC into a two level optimization problem in which the upper level problem finds the optimal platform configuration while each subproblem optimizes the individual products. The proposed approach was demonstrated in optimizing a family of ten bathroom scales. The Pareto front reveals the trade-off between commonality and the ability to achieve distinct performance targets, which can help in product family planning. Moreover, comparing the optimal fronts using the ECI versus the all-or-none metric for commonality revealed the importance of generalizing the commonality definition to allow component sharing among subsets of variants in the optimization formulation, and the computational studies demonstrate the advantages of decomposition.

In conclusion, since the all-or-none commonality restriction has been shown to generate inferior solutions, it should only be used when generalized commonality is truly not of interest or when specific knowledge about the product domain reveals that all-or-none commonality is sufficient. First, using the restriction as a computational convenience can result in significant losses. Second, using a suitable definition of the commonality objective is also important to provide an appropriate proxy for commonality benefits, and we advocate for use of the extended CI. Finally, restrictions of the joint product platform selection and variant design problem to a priori fixed platform specifications and solution methods that rely on multiple stages can generate suboptimal solutions. The proposed method addresses the joint problem directly, avoiding these limitations, and the decomposition scheme addresses the scalability issue and enables use of parallel processing to further speed up computation. However, in order to obtain more concrete conclusions regarding the performance of various existing product family optimization methods, a comprehensive and systematic comparison study for solving a number of different case studies using all proposed approaches is required.

To date, no method has been proposed with proven achievement of global optimality for the joint product family problem; however, we believe that efficient approaches to address the joint problem directly while also accounting for generalized commonality is an important direction for future research.

Acknowledgment

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] Thevenot, H. J., and Simpson, T. W., 2006, "Commonality Indices for Product Family Design: A Detailed Comparison," J. Eng. Design, **17**(2), pp. 99–119.
[3] Collier, D. A., 1981, "The Measurement and Operating Benefits of Component Part Commonality," Decision Sci., **12**(1), pp. 85–96.
[4] Martin, M. V., and Ishii, K., 2002, "Design for Variety: Developing Standardized and Modularized Product Platform Architectures," Res. Eng. Des., **13**(4), pp. 213–235.
[5] Wacker, J. G., and Trelevan, M., 1986, "Component Part Standardization: An Analysis of Commonality Sources and Indices," J. Operations Manage., **6**(2), pp. 219–240.
[6] Jiao, J., and Tseng, M. M., 2000, "Understanding Product Family for Mass Customization by Developing Commonality Indices," J. Eng. Design, **11**(3), pp. 225–243.
[7] Kota, S., and Sethuraman, K., 1998, "Managing Variety in Product Families Through Design for Commonality," *Design Theory and Methodology—DTM'98*, Atlanta, GA, ASME Paper No. DETC98/DTM-5651.
[8] Siddique, Z., Rosen, D. W., and Wang, N., 1998, "On the Applicability of Product Variety Design Concepts to Automotive Platform Commonality," *ASME Design Engineering Technical Conferences—Design Theory and Methodology*, Atlanta, GA, ASME Paper No. DETC98/DTM-5661.
[9] Thevenot, H. J., and Simpson, T. W., 2005, "A Comprehensive Metric for Evaluating Commonality in a Product Family," J. Eng. Design, **18**(6), pp. 577–598.
[10] 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, ASME.
[11] Hassan, R., De Weck, O., and Springmann, P., 2004, "Architecting a Communication Satellite Product Line," *22nd AIAA International Communications Satellite Systems Conference & Exhibit 2004 (ICSSC)*, Monterey, CA, AIAA.
[12] Khire, R. A., Messac, A., and Simpson, T. W., 2006, "Optimal Design of Product Families Using Selection-Integrated Optimization (SIO) Methodology," *11th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Portsmouth, VA, AIAA.
[13] Messac, A., Martinez, M. P., and Simpson, T. W., 2002, "A Penalty Function for Product Family Design Using Physical Programming," ASME J. Mech. Des., **124**(2), pp. 164–172.
[14] Nayak, R. U., Chen, W., and Simpson, T. W., 2002, "A Variation-Based Method for Product Family Design," Eng. Optimiz., **34**(1), pp. 65–81.
[15] Simpson, T. W., and D'Souza, B. S., 2004, "Assessing variable levels of platform commonality within a product family using a multiobjective genetic algorithm," Concurr. Eng. Res. Appl., **12**(2), pp. 119–129.
[16] Fellini, R., Kokkolaras, M., Papalambros, P., and Perez-Duarte, A., 2005, "Platform Selection Under Performance Loss Constraints in Optimal Design of Product Families," ASME J. Mech. Des., **127**(4), pp. 524–535.
[17] Cetin, O. L., and Saitou, K., 2004, "Decomposition-Based Assembly Synthesis for Structural Modularity," ASME J. Mech. Des., **126**(2), pp. 234–243.
[18] 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, ASME.
[19] 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, ASME.
[20] Khajavirad, A., Michalek, J. J., and Simpson, T. W., 2007, "A Decomposed Genetic Algorithm for Solving the Joint Product Family Optimization Problem," *Third AIAA Multidisciplinary Design Optimization Specialists Conference*, Honolulu, HI, AIAA.
[21] Kim, H. M., Chen, W., and Wiecek, M. M., 2006, "Lagrangian coordination for Enhancing the Convergence of Analytical Target Cascading," AIAA J., **44**(10), pp. 2197–2207.
[22] Michalek, J. J., and Papalambros, P. Y., 2005, "An Efficient Weighting Update Method to Achieve Acceptable Consistency Deviation in Analytical Target Cascading," ASME J. Mech. Des., **127**(2), pp. 206–214.
[23] Michelena, N., Park, H., and Papalambros, P. Y., 2003, "Convergence Properties of Analytical Target Cascading," AIAA J., **41**(5), pp. 897–905.
[24] Tosserams, S., Etman, L. F. P., Papalambros, P. Y., and Rooda, J. E., 2006, "An Augmented Lagrangian Relaxation for Analytical Target Cascading Using the Alternating Direction Method of Multipliers," Struct. Multidiscip. Optim., **31**(3), pp. 176–189.
[25] Li, Y., Lu, Z., and Michalek, J. J., 2008, "Diagonal Quadratic Approximation for Parallelization of Analytical Target Cascading," ASME J. Mech. Des., **130**(5), p. 051402.
[26] Choudhary, R., Malkawi, A., and Papalambros, P. Y., 2005, "Analytic Target Cascading in Simulation-Based Building Design," Autom. Constr., **14**(4), pp. 551–568.
[27] Kim, H. M., Rideout, D. G., Papalambros, P. Y., and Stein, J. L., 2003, "Analytical Target Cascading in Automotive Vehicle Design," ASME J. Mech. Des., **125**(3), pp. 481–489.
[28] Kokkolaras, M., Fellini, R., Kim, H. M., Michelena, N. F., and Papalambros, P. Y., 2002, "Extension of the Target Cascading Formulation to the Design of Product Families," Struct. Multidiscip. Optim., **24**(4), pp. 293–301.
[29] Michalek, J. J., Feinberg, F. M., and Papalambros, P. Y., 2005, "Linking Marketing and Engineering Product Design Decisions via Analytical Target Cascading," Journal of Product Innovation Management, **22**, pp. 42–62.
[30] Michalek, J. J., Ceryan, O., Papalambros, P. Y., and Koren, Y., 2006, "Balancing Marketing and Manufacturing Objectives in Product Line Design," ASME J. Mech. Des., **128**(6), pp. 1196–1204.
[31] Kumar, D., Chen, W., and Simpson, T., 2007, "A Market-Driven Approach to the Design of Platform-Based Product Families," Int. J. Prod. Res., in press, published online August 2007.