Comprehensive Product Platform Planning (CP^3) Framework

Souma Chowdhury
Doctoral Student
Student Member ASME
Department of Mechanical, Aerospace, and Nuclear Engineering,
Multidisciplinary Design and Optimization Laboratory (MDOL),
Rensselaer Polytechnic Institute,
Troy, NY 12180
e-mail: chowds@rpi.edu

Achille Messac[1]
Distinguished Professor and Department Chair
Fellow ASME
Department of Mechanical and Aerospace Engineering,
Syracuse University,
Syracuse, NY 13244
e-mail: messac@syr.edu

Ritesh A Khire
Senior Research Engineer
Mem. ASME
United Technologies Research Center (UTRC),
East Hartford, CT 06118
e-mail: KhireR@utrc.utc.com

Development of a family of products that satisfies different market niches introduces significant challenges to today's manufacturing industries—from development time to aftermarket services. A product family with a common platform paradigm offers a powerful solution to these daunting challenges. This paper presents a new approach, the Comprehensive Product Platform Planning (CP^3) *framework, to design optimal product platforms. The* CP^3 *framework formulates a generalized mathematical model for the complex platform planning process. This model (i) is independent of the solution strategy, (ii) allows the formation of sub-families of products, (iii) allows the simultaneous identification of platform design variables and the determination of the corresponding variable values, and (iv) seeks to avoid traditional distinctions between modular and scalable product families from the optimization standpoint. The* CP^3 *model yields a mixed integer nonlinear programming problem, which is carefully reformulated to allow for the application of continuous optimization using a novel Platform Segregating Mapping Function (PSMF). The PSMF can be employed using any standard global optimization methodology (hence not restrictive); particle swarm optimization has been used in this paper. A preliminary cost function is developed to represent the cost of a product family as a function of the number of products manufactured and the commonality among these products. The proposed* CP^3 *framework is successfully implemented on a family of universal electric motors. Key observations are made regarding the sensitivity of the optimized product platform to the intended production volume.*
[DOI: 10.1115/1.4004969]

Keywords: commonality matrix, MINLP, optimization, particle swarm, platform, product family

1 Introduction

A product family consists of a set of products that share certain common features that are embodied in a platform. Different products within the family are produced by customizing specific additional features on the platform. By doing so, a group of related products can be derived from a common product platform to satisfy a variety of market niches. Also, the sharing of a common platform by different products is expected to result in: (i) reduced overhead, (ii) lower per-product cost, and (iii) increased profit. A key feature affecting the success of a product family is the effectiveness of the product platform around which the product family is derived. By sharing components and production processes across a platform of products, companies can develop different products efficiently. This approach also increases the flexibility and the responsiveness of their manufacturing processes, and is likely to take away market share from competitors that develop one product at a time. In addition, the domain of product platform planning can be extended to multiple sets/series of products, thereby providing the flexibility of creating sub-families of products as well. In this section, we briefly discuss the state of the art in product family design; this discussion is followed by a survey of the techniques (and their limitations) reported in the literature to solve different types of product families. The section ends with an introduction to the product platform planning framework developed in this paper.

1.1 Product Family Design. In the field of management, product platform research is often addressed in terms of qualitative product planning problems [1,2]. The qualitative product planning tools typically support managerial and strategic decision making. However, they do not offer practical support to system-level designers, who must develop a product architecture to deliver the different products, while sharing parts and production steps across the products [3]. In particular, designers must translate the qualitative leveraging strategies into useful customer requirements to guide platform-based product development. These issues are addressed in the engineering design domain, and employ quantitative methods. A characteristic distinction between management and engineering approaches is that the former typically rely on conceptual and/or qualitative methods, while the latter typically rely on quantitative analysis/optimization methodologies.

Various optimization-based product platform planning approaches and formulations have been proposed in the literature, which includes the methods summarized by Simpson et al. [4]. Several other approaches have been proposed in recent years, 2006–2010. These papers often present diverse objectives and initial assumptions that may not readily apply to a broader scenario. Little work has been done to develop a formal framework to coherently address different problem scenarios. In this paper, the new Comprehensive Product Platform Planning (CP^3) framework is presented to address these issues from a broad perspective. The proposed CP^3 framework simultaneously delivers an encompassing product family model and a generic design optimization methodology, which can address the different classes of problems. The complete process of commercial product family development requires the adequate consideration of and the balancing of the engineering and marketing objectives [5]. The proposed CP^3 framework is developed as a robust strategy to achieve the engineering design objective of product platform planning.

Depending on their design, product families have traditionally been classified as (1) modular (module-based), or (2) scalable

[1]Corresponding author.

Contributed by the Design Theory and Methodology Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received January 6, 2011; final manuscript received August 16, 2011; published online October 18, 2011. Assoc. Editor: Janet K. Allen.

(scale-based) families. In a scale-based product family, each individual product is comprised of the same set of physical design variables. Different products in the family are developed by scaling the nonplatform features (design variable) such that each product satisfies a unique set of requirements. In a module-based product family, distinct modules are added or substituted (to a common platform) to develop different products [6,7]. A popular example of a modular product family is the series of Sony Walkmans [8,9], whereas a standard example of a scalable product family is Boeing's 777 aircraft series [10]. Major automobile manufacturers have also made efforts toward the use of scalable product families [11].

1.2 Existing Research in Product Family Design (PFD)

1.2.1 Scale-Based Product Families. Under prevailing approaches to the design optimization of scale-based product families, two critical decisions typically made are: (i) the selection of platform and scaling design variables and (ii) the determination of the values of these design variables. The selection of platform and scaling design variables is combinatorial in nature; while determining the variable values is generally continuous in nature. The combinatorial aspect makes the design optimization of product families a challenging one [12]. In the literature, researchers have proposed two major classes of methodologies to address these challenges: (i) the two-step approach and (ii) the exhaustive approach.

A comprehensive list of different "two-step" methods can be found in the book chapter by Simpson [13]. In the two-step class of methods, the selection of platform and scaling design variables is performed separately from the optimization of the product family. This technique is relatively simple to implement, and generally does not demand extensive computational resources. However, the two step strategy potentially introduces a significant source of sub-optimality. In the exhaustive approach, all the possible combinations of platform and scaling design variables are individually optimized and mutually compared, to yield the overall optimum. This approach can guarantee (assuming the correctness of the individual optimizations) the overall optimum product family design. On the negative side, the methods based on the exhaustive approach are likely to become computationally prohibitive for systems with a large number of design variables—a n-variable system would lead to 2^n possible combinations of platform and scaling design variables (each representing a unique family). If the creation of sub-families is also considered, the number of possible combinations is exponentially higher (2^{rn}, where $r \geq 1$). Detailed discussion of the dimensionality of platform-scaling combinations can be found in the paper by Chowdhury et al. [14]. The processes of a typical two-step approach and a typical exhaustive approach are depicted in Figs. 1(*a*) and 1(*b*), respectively.

In recent years, new product family design approaches have been proposed, which can be applied to a wider array of scaled based product families; these methods address most of the limitations of the earlier methods, and also present other uniquely favorable characteristics. The selection integrated optimization (SIO) approach introduced by Khire et al. [15,16] addresses the sub-optimality of two-step methods by integrating the (i) platform identification process and the (ii) product family optimization process. A new variable segregating mapping function [15,17] converts the discrete combinatorial process (of platform identification) into a continuous process. This approach provides a robust and computationally inexpensive (compared with other single stage methods) product family optimization framework.

Another new class of PFD methods uses multi-objective genetic algorithms to design a product family. A single stage approach (without a priori platform identification), based on a decomposition solution strategy that uses the binary Non-Sorting Genetic Algorithm-II (NSGA-II), is presented by Khajavirad et al. [18]. This method provides flexibility in allowing the formation of a platform, whenever a design variable (value) is shared by more than one product, and not necessarily all products in the family, which eliminates the *all-common or all-distinct restriction*. The significant computational expense of the binary GA approach, especially in the case of large scale problems, is addressed using a parallelized sub-GA solution strategy. Similar flexibility in platform creation is also presented by Chen and Wang [19]. The latter paper presents a PFD method that uses a 2-Level Chromosome Genetic algorithm (2-LCGA), and proposes an *information theoretical approach* that incorporates fuzzy clustering and Shannon's entropy to identify platform design variables. The process of platform creation precedes performance optimization of the product family. Consequently, the method developed by Chen et al. [19] exhibits the limiting attributes of the "two-step" approach. Scale-based PFD methodologies that use physical programming [20,21], and that explore the impact of design variable uncertainty [22], have also been reported in the literature.

1.2.2 Module-Based Product Families. One of the popular approaches to module-based product family design conceptually divides the process into the following three levels (i) Architectural level: to establish a system structure and its variations, (ii) Configuration level: to establish standard configuration(s), and it's variations of products and modules, and (iii) Instantiation level: to develop a practical product family through variable quantification and combinatorial selection of the modules. In this paper, the instantiation level of modular PFD has been particularly addressed. Further investigation needs to be performed to expand the scope of the CP^3 framework to the broad spectrum of modular product family design. The instantiation task level is composed of the following two phases (i) Variable quantification: to develop modules across product prototypes by quantifying design variables and (ii) Combinatorial selection: to develop product prototypes by selecting desirable combinations from the feasible ones. Based on these phases, approaches to the instantiation task level can be divided into the following three classes:

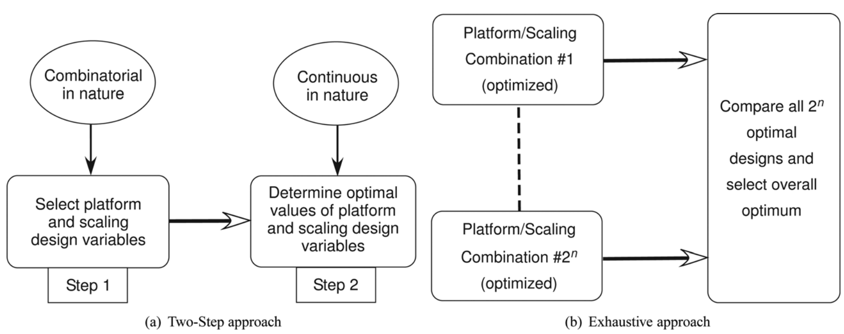

Fig. 1 Existing methods to design scale-based product families

(1) Optimization of module attributes under fixed module combination;
(2) Optimization of module combinations using predefined module candidates; and
(3) Simultaneous optimization of module attributes and module combinations.

The majority of the approaches for solving these optimization problems require specifying the platform (fixed module combination, i.e., Class 1) prior to optimization, in order to reduce the design space and render the problem computationally more tractable. Most other optimization approaches are geared toward Class 2 optimization problems, e.g., Ref. [23]. The assumptions involved in these two classes may lead to sub-optimal module-based product families. Very few optimization approaches exist to solve Class 3 type optimization problems, such as developed by Fujita and Yoshida [24,25].

Several other well known methods exist in modular PFD, such as presented by Stone et al. [26], Dahmus et al. [27], Guo and Gershenson [28], Jose and Tollenaere [29], Kalligeros et al. [30], Saron et al. [31], and Yu et al. [32]. Detailed discussion of these methodologies is not within the scope of this paper.

1.2.3 Limitations of the Existing PFD Methods. The scope of application of the earlier PFD methods [13,15] is restricted by the assumption—"a platform is formed only when the value of a design variable is shared by all product variants in the family." This assumption, which is termed as the *all-common or all-distinct restriction* hinders the creation of sub-families of products, unless additional prior design assumptions are made. Most importantly, the majority of the existing PFD methodologies (including recent methods [18,19]) do not readily represent the platform planning process by *a generalized mathematical model*; as a result, only a particular class of optimization algorithms can be leveraged in these methods, which is restrictive. Moreover, a generalized mathematical model can facilitate helpful investigation of the underlying mathematics (complex and combinatorial) of product platform planning, which has rarely been offered by the existing PFD methodologies.

The scale-based PFD methods assume that each product is comprised of all the design variables involved in the family; as a result, these methods do not readily apply to modular product families. On the other hand, typical modular PFD methods often cannot readily account (without platform/scaling assumptions) for the simultaneous presence of modular and scalable product attributes: e.g., in a laptop, the DVD-drive can have both modular (DVD, HD-DVD, and Blue-Ray-DVD) and scalable (drive speeds $= 1\times,\ 2\times,\ldots, N\times 1.35$ MB/s) properties. In addition to these limitations, most of the existing PFD approaches use a penalty function or a commonality metric to account for the reduction in manufacturing costs, resulting from platform planning. These artificial cost functions are usually determined (i) using the differences in the magnitude of each design variable across different products [15] or (ii) using the quantified commonalities (e.g., a commonality index) among products [18]. Such artificial cost functions are often limited by the inherent assumptions that are rarely stated explicitly—e.g., *the cost reduction that results from creating product platforms is independent of the "intended production volume" for each product variant.* The CP^3 framework seeks to address these limitations, further discussion of which is given in Sec. 1.3.

1.3 Comprehensive Product Platform Planning (CP^3) Framework. Through the CP^3 framework, the objective is to address the critical attributes of product family design. This framework consists of two components: (i) a comprehensive and flexible product family model that yields a mixed integer nonlinear programming (MINLP) problem and (ii) a design optimization strategy to solve this MINLP problem. In the remainder of this paper, these components will be referred to as the "CP^3 Model" and the "CP^3 Optimization", respectively. The CP^3 framework, presented in this paper, has the flexibility to design a majority of the scale-based product families that occur in practice. Additionally, the CP^3 framework, in its current form, also seeks to address the instantiation level (of the three level modular PFD approach) of module-based product family design.

The principal contributions of the CP^3 model are:

(1) This model presents a generalized and compact mathematical representation of the platform planning process, which is independent of the optimization strategy.
(2) This model avoids the *all-common/all-distinct restriction* [18], thereby allowing the formation of sub-families.
(3) This model facilitates simultaneous (i) selection of platform/scaling design variables and (ii) quantification of the optimal design variable values.

The CP^3 model formulates a generic equality constraint (the *commonality constraint*) to represent the variable based platform formation process. The presence of a combination of binary variables and continuous variables is attributed to the combinatorial process of platform identification. The nonlinearity of this problem can be primarily attributed to the likely nonlinear nature of the system model for the products: introduced by typical nonlinear performance function and nonlinear system constraints. To the best of the authors' knowledge, such a generalized and closed form mathematical representation of the platform planning process is unique in the literature.

The CP^3 optimization strategy introduces the platform segregating mapping function (*PSMF*) to solve the MINLP problem at a much reduced computational expense. In the case of a generic product family, comprising N products and a total of n design variables for each product, the resulting MINLP problem contains at least "$n \times {}^{N}C_2 = n \times N(N-1)/2$" binary-integer variables. In the CP^3 model, these binary-integer variables determine the commonalities across products with respect to particular physical design variables. Typical gradient based MINLP solvers (e.g., branch and bound, cutting plane, and outer approximation algorithms [33,34]) may not be effective at solving such a complex and highly nonlinear optimization problem. The *PSMF* approach is inspired by the concept of converting the mixed integer problem into a continuous variable problem, as developed in the selection integrated optimization method [15]. The PSMF technique also exhibits a strong potential for application to the design of adaptive systems—adaptive systems are engineering systems that are required to maintain high performance under changing environments [16]. The dependence of the manufacturing cost of a product on the number of similar products (products sharing design variable values) is expressed using a cost decay function (*CDF*).

Particle swarm optimization (PSO) algorithm [35] is used to solve the approximated MINLP problem. A robust constraint handling technique, introduced by Deb et al. [36], and later adopted by Chowdhury and Dulikravich [37], is employed to deal with the constraints involved in the optimization problem. A series of numerical experiments indicated that the approximated MINLP problem is most likely to be multimodal, which can be attributed to the formulated *commonality constraint*. PSO, being a stochastic search algorithm, deals with multimodal problems significantly better than do gradient based algorithms. Moreover, PSO is straight forward to implement and involves fewer user defined parameters that need to be adjusted, when compared to some of the standard evolutionary optimization algorithms.

The CP^3 framework is expected to be particularly more effective in the case of product families (i) that have sub-families of products, and/or (ii) that involve a combination of modular and scalable product attributes. Specific examples of such families include consumer electronics products, such as **laptops**. Such features as display size, hard-disk-drive capacity, size of RAM,

Table 1 List of variables for the product family example

Physical design variables	Product 1	Product 2	Integer variables
1st variable	x_1^1	x_1^2	λ_1^{12}
2nd variable	x_2^1	x_2^2	λ_2^{12}
3rd variable	x_3^1	x_3^2	λ_3^{12}

and processor speed are typically scalable attributes, whereas web-camera, microphone, speakers, and optical-drive are typically modular attributes. In addition, a single manufacturer often provides several laptop brands/series (e.g., Dell-Vostro, Dell-Inspiron, and Dell-XPS series) that includes further model variants (e.g., XPS 17″ 3D, XPS 17″, and XPS 15″); individual models might have commonalities within the series, and also across different series. Upfront decision making at the engineering design/development stage, regarding how the product-line can be divided into sub-families, can be challenging. Similarly, the classification of a product family as strictly modular or scalable at the design stage introduces limiting assumptions, and requires the design engineer to seek methods that can be effectively leveraged under those specific scenarios. The CP^3 framework provides the flexibility to design sub-families, while simultaneously addressing modular and scalable attributes in a product family. Thereby, this method provides a much desirable one-step solution to "product family design engineers". To summarize, the properties of the comprehensive product family model presented in this paper are not only mathematically interesting but also attractive to the engineering practice in an industrial setting.

In Sec. 2, the Comprehensive Product Platform Planning (CP^3) model is developed. In Sec. 3, the cost objective for the product family is formulated. Section 4 discusses the CP^3 optimization strategy that includes the developments of the platform segregating mapping function and the cost decay function. A brief description of the universal electric motor and the design optimization of a family of motors using the CP^3 technique are presented in Sec. 5. Concluding remarks are provided in the last section (Sec. 6).

2 Comprehensive Product Platform Planning (CP^3) Model

In this section, we develop a product platform planning model that is both comprehensive and independent of the eventual solution strategy. This model not only allows effective design optimization of product families but also promotes further investigation of the underlying mathematical processes in product platform planning. The formulation of the CP^3 Model is followed by an illustration of this model using the example of a representative "4 products - 5 variables" family. This section is concluded with the definition of the overall optimization problem yielded by the CP^3 Model.

Table 2 Product platform plan for the "4 products – 5 variables" sample family

Physical design variables	Product 1	Product 2	Product 3	Product 4
1st variable (x_1)	◇	◇	◇	◇
2nd variable (x_2)	—	—	—	—
3rd variable (x_3)	□	□	O	O
4th variable (x_4)	—	—	—	—
5th variable (x_5)	△	—	—	△

2.1 Formulation of the CP^3 Model.

The CP^3 Model is founded on the concept of a mixed-integer nonlinear equality constraint. This formulation is illustrated using a representative family of two products that are comprised of three design variables each. Table 1 shows the variables involved in this example.

For a design variable x_j^k, in Table 1, the superscript (k) and the subscript (j) represent the product number and the variable number, respectively. The binary-integer variables (λ variables), given in the table, are defined as

$$\lambda_j^{12} = \begin{cases} 1, & \text{if } x_j^1 = x_j^2 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Hence, the λ variables determine the commonality among products, with respect to the physical design variables. The general MINLP problem, formulated to represent the design optimization of the product family example (shown in Table 1), is given by

$$\begin{aligned}
&\text{Max} \quad f_{\text{perf}}(Y)\\
&\text{Min} \quad f_{\text{cost}}(Y)\\
&\text{subject to}\\
&\lambda_1^{12}\left(x_1^1 - x_1^2\right)^2 + \lambda_2^{12}\left(x_2^1 - x_2^2\right)^2 + \lambda_3^{12}\left(x_3^1 - x_3^2\right)^2 = 0\\
&g_i(X) \le 0, \quad i = 1, 2, \ldots, p\\
&h_i(X) = 0, \quad i = 1, 2, \ldots, q\\
&\text{where}\\
&Y = \left\{x_1^1, x_2^1, x_3^1, x_1^2, x_2^2, x_3^2, \lambda_1^{12}, \lambda_2^{12}, \lambda_3^{12}\right\}\\
&X = \left\{x_1^1, x_2^1, x_3^1, x_1^2, x_2^2, x_3^2\right\}\\
&\left(\lambda_1^{12}, \lambda_2^{12}, \lambda_3^{12}\right) \in B : B = \{0, 1\}
\end{aligned} \tag{2}$$

and where f_{perf} and f_{cost} are the objective functions that represent the performance and the cost of the product family, respectively. In Eq. (2), g_i and h_i represent the inequality and equality constraints related to the physical design of the product. The first equality constraint in Eq. (2), which involves the parameters λ_j^{ik} is termed the *commonality constraint*, which can be represented in a compact matrix format as

$$\begin{bmatrix} x_1^1 & x_1^2 & x_2^1 & x_2^2 & x_3^1 & x_3^2 \end{bmatrix} \times \begin{bmatrix} \lambda_1^{11} & -\lambda_1^{12} & 0 & 0 & 0 & 0 \\ -\lambda_1^{21} & \lambda_1^{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_2^{11} & -\lambda_2^{12} & 0 & 0 \\ 0 & 0 & -\lambda_2^{21} & \lambda_2^{22} & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_3^{11} & -\lambda_3^{12} \\ 0 & 0 & 0 & 0 & -\lambda_3^{21} & \lambda_3^{22} \end{bmatrix} \begin{bmatrix} x_1^1 \\ x_1^2 \\ x_2^1 \\ x_2^2 \\ x_3^1 \\ x_3^2 \end{bmatrix} = 0 \tag{3}$$

This formulation can be extended to a general product family, comprising N products and n design variables, as given by

Table 3 Definition of the platform symbols for the "4 products – 5 variables" sample family

Platform-1	Platform-2	Platform-3	Platform 4
◇	□	O	△

$X^T\Lambda X=0$, where

$$\Lambda=\begin{bmatrix}
\sum\limits_{k\neq 1}\lambda_1^{1k} & \cdots & -\lambda_1^{1N} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\vdots & \vdots & \vdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-\lambda_1^{N1} & \cdots & \sum\limits_{k\neq N}\lambda_1^{Nk} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & \sum\limits_{k\neq 1}\lambda_j^{1k} & \cdots & -\lambda_j^{1N} & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & -\lambda_j^{N1} & \cdots & \sum\limits_{k\neq N}\lambda_j^{Nk} & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sum\limits_{k\neq 1}\lambda_n^{1k} & \cdots & -\lambda_n^{1N}\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \vdots & \vdots & \vdots\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_n^{N1} & \cdots & \sum\limits_{k\neq N}\lambda_n^{Nk}
\end{bmatrix} \tag{4}$$

where $k=1,2,.....,N;\ \ j=1,2,.....,n;$

$$X=\begin{bmatrix}x_1^1 & x_1^2 & \cdots & x_1^N & \cdots & x_j^1 & x_j^2 & \cdots & x_j^N & \cdots & x_n^1 & x_n^2 & \cdots & x_n^N\end{bmatrix}^T$$

The matrix Λ is called the *commonality constraint matrix*. This matrix is a symmetric block diagonal matrix, where the jth block corresponds to the jth design variable. An explicit representation of each block of the Λ matrix is given by

$$\Lambda_j=\begin{bmatrix}
\sum\limits_{k\neq 1}\lambda_j^{1k} & -\lambda_j^{12} & \cdots & -\lambda_j^{1l} & \cdots & -\lambda_j^{1N}\\
-\lambda_j^{21} & \sum\limits_{k\neq 2}\lambda_j^{2k} & \cdots & -\lambda_j^{2l} & \cdots & -\lambda_j^{2N}\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
-\lambda_j^{l1} & -\lambda_j^{l2} & \cdots & \sum\limits_{k\neq l}\lambda_j^{lk} & \cdots & -\lambda_j^{lN}\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
-\lambda_j^{N1} & -\lambda_j^{N2} & \cdots & -\lambda_j^{Nl} & \cdots & \sum\limits_{k\neq N}\lambda_j^{Nk}
\end{bmatrix} \tag{5}$$

The *commonality constraint matrix* can be derived from the generalized *commonality matrix* λ that is expressed as

$$\lambda=\begin{bmatrix}
\lambda_1^{11} & \cdots & \lambda_1^{1N} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\vdots & \vdots & \vdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\lambda_1^{N1} & \cdots & \lambda_1^{NN} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & \lambda_j^{11} & \cdots & \lambda_j^{1N} & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & \lambda_j^{N1} & \cdots & \lambda_j^{NN} & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_n^{11} & \cdots & \lambda_n^{1N}\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \vdots & \vdots & \vdots\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_n^{N1} & \cdots & \lambda_n^{NN}
\end{bmatrix} \tag{6}$$

where

$$\lambda_j^{kl}=\left\{\begin{array}{ll}1, & \text{if } \lambda_j^{kk}=\lambda_j^{ll}=1 \text{ and } x_j^l=x_j^k\\ 0, & \text{otherwise}\end{array}\right\}\forall k\neq l$$

$$\lambda_j^{kk}=\begin{cases}1, & \text{if the } j\text{th variable is included in product}-k\\ 0, & \text{if the } j\text{th variable is NOT included inproduct}-k\end{cases}$$

$k,l=1,2,.....,N;\quad j=1,2,.....,n;$

It can be observed from Eq. (6) that the *commonality matrix* is also a symmetric block diagonal matrix. The off-diagonal elements of the *commonality matrix* (λ_j^{kl}) will be referred to as the *commonality variables*. The diagonal elements of the *commonality matrix* (λ_j^{kk}) determine whether the jth variable is included in product-k. This *commonality matrix* definition is somewhat similar to that reported by Khajavirad and Michalek [38]. Every block of the *constraint commonality matrix* can be expressed as a function of the corresponding *commonality matrix* block, which is

$$\Lambda_j = f_{con}(\lambda_j) \tag{7}$$

In typically modular product families, different products might comprise different types and different numbers of modules. Each module is comprised of a particular set of design variables that are also known as module attributes; these attributes may be shared by more than one module in a complex system. Consequently, different products can be comprised of physically different sets of design variables. In order to address a modular product family design, a product platform planning model should account for the inclusion, the exclusion and the substitution of design variables. These three possibilities can be captured by the novel *commonality matrix*. In this context, one of the following three distinct scenarios can occur:

(1) The jth design variable **is required** in product-k; consequently, $\lambda_j^{kk} = 1$ would be known a priori (fixed).
(2) The jth design variable **is not relevant** for product-k; consequently, $\lambda_j^{kk} = 0$ would be known a priori (fixed).
(3) Inclusion of the jth design variable **is optional** for product-k; the corresponding λ_j^{kk} variable is allowed to be determined during the course of product family optimization (variable).

If the third scenario occurs, the CP^3 model does not make any prior assumptions regarding the attribute values or whether a module (or the corresponding attributes) will be shared by multiple products or will not form any platform. If the first scenario occurs, the module attributes themselves may scale among the products that must the concerned module. The *commonality matrix* blocks representing the scaling design variables (attributes included in all product variants) should have the diagonal elements fixed at one. Therefore, careful specification/management of the diagonal elements of the commonality matrix automatically addresses the modularity/scalability properties of the design variables, without imposing limiting distinctions. This model allows a coherent consideration of the likely combination of scaling and modular attributes in a product family, which is uniquely helpful.

The helpful capability of the CP^3 model to flexibly address combined modular-scaling families is further explained using the example of a *family of laptop PCs*—Three laptops (a 15 in. notebook, a 13 in. notebook, and a netbook) are being designed, and the manufacturer decides that the 15 inch notebook **will** have a DVD-drive, the 13 inch notebook **might/might not** have a DVD-drive (based on the ensuing overall product cost), and the netbook **will not** have a DVD-drive (due to space constraints). A key constituent design variable of the DVD-drive module is the drive-speed/data-rate (in "$\times$ 1.35MB/s") that is a scaling attribute: e.g., "1 $\times$ or 2 $\times$ 1.35 MB/s". In this case, the "3 $\times$ 3" commonality matrix block corresponding to this design variable (drive speed/data-rate) can be expressed as

$$\lambda_{\text{DVD}} = \begin{bmatrix} 1 & \lambda^{12} & 0 \\ \lambda^{21} & \lambda^{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8}$$

where the order of the products is [15 in. notebook, 13 in. notebook, and the netbook], and the commonality variables λ^{12}, λ^{21}, and λ^{22} are to be determined during optimization. The *commonality matrix* formulation presented in this paper thus provides a generalized product platform planning model, which can address a wide variety of product families without making limiting distinctions between scalable and modular attributes.

It is helpful to note that, factors such as (i) the product-module architecture and (ii) the availability of module options in a modular family demand additional considerations, which are not explicitly addressed by the current CP^3 model. For example, in the case of laptops, technologically differing/substitutable types of DVD drives are commercially available: e.g., DVD-R, HD-DVD, and Blue-Ray-drive. Although, these DVD-drive options have the same drive-speed attribute, the commonality representation in the CP^3 model would not readily apply if two laptop variants were to use two different DVD-drive types. Moreover, the modules in a product are often not independent of each other from a design and/or operation perspective; in that case, inclusions/exclusion/substitution of modules can be mutually dependent (e.g., web-camera and microphone), which leads to dependent commonality matrix blocks. Such "*module interdependency*" that can be defined within the conceived product architecture is not addressed by the current CP^3 model. Appropriate considerations of the underlying product architecture and module options should further advance the applicability of the CP^3 model, and is considered a key area for future research.

2.2 Representative Illustration of the CP^3 Model: A "4 Products – 5 Variables" Family.

The proposed CP^3 model is illustrated using an example of a product family comprising 4 products. It is helpful at this point to provide a careful definition of a generalized product platform—"*A product platform is said to be created when more than one product variant in a family share a particular design variable*." In this case "sharing a design variable between two products" implies that the products are designed to have the same value of the concerned variable. Table 2 shows the platform plan of the sample product family. Each symbol in Table 2, except the "–" symbol, represents a platform. Blocks in Table 1, displaying similar symbols, imply that the corresponding products are members of a particular variable-based platform. A block, displaying the "–" symbol, represents a non-platform (scaling) design variable value—which implies the corresponding design variable value is not shared by more than one product. Blocks without any specified symbol (blank) imply that the particular variable is not included in the corresponding product. The platform symbols are defined in Table 3.

A product platform plan, as shown in the example (in Table 2), entails classifying design variables (in the entire family) into the following three categories:

(1) *Platform design variable*: A design variable that is shared by all the different kinds of products in the family; e.g., variable x_1 in Table 2.
(2) *Sub-platform design variable*: A design variable that is shared by a particular set of product variants in the family, leading to a sub-family; e.g., variable x_5 in Table 2. Sub-platforms may also lead to multiple sub-families with respect to a design variable; e.g., variable x_3 in Table 2.
(3) *Nonplatform design variable*: A design variable that is not shared by more than one product in the family; e.g., variables x_2 and x_4 in Table 2.

The diagonal blocks of the *commonality matrix*, corresponding to each design variable for the product family illustrated in Table 2, is given by

$$\lambda_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad \lambda_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\lambda_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad \lambda_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{9}$$

$$\lambda_5 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

The corresponding five diagonal blocks of the *constraint commonality matrix* (Λ), determined from the *commonality matrix*, are given by

$$\Lambda_1 = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}, \quad \Lambda_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\Lambda_3 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \quad \Lambda_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{10}$$

$$\Lambda_5 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

2.3 The Generalized MINLP Problem. The generalized MINLP problem for a family of N products, comprising a global set of n design variables, can be expressed as

$$\begin{aligned} &\text{Max } f_{perf}\ (Y) \\ &\text{Min } f_{\cos t}\ (Y) \\ &\quad \text{subject to} \\ &X^T \Lambda X = 0 \\ &g_i(X) \le 0, \quad i = 1, 2, \ldots, p \\ &h_i(X) = 0, \quad i = 1, 2, \ldots, q \\ &\quad \text{where} \\ &\Lambda = f_{con}(\lambda) \\ &Y = \{X, \lambda\} \\ &X = \left[x_1^1 x_1^2 \ldots\ x_1^N \ldots\ x_j^1 x_j^2 \ldots\ x_j^N \ldots\ x_n^1 x_n^2 \ldots\ x_n^N\right]^T \\ &\lambda_j^{lk} \in B : B = \{0, 1\} \\ &k, l = 1, 2 \ldots, N;\ j = 1, 2 \ldots, n; \end{aligned} \tag{11}$$

and where the matrices Λ and λ are given by Eqs. (4) and (6), respectively. It is helpful to note that, although the matrix λ is a variable for the MINLP problem, certain diagonal elements (λ_j^{kk}) are known apriori (Refer Sec. 2).

3 Cost Objective for a Product Family

In this section, a preliminary cost analysis of a product family is performed to express the cost as a function of factors that are considered to be critical to platform planning. We do not intend to develop a detailed cost model of a product family; the cost of a commercial product family would be significantly sensitive to the actual product line being planned/designed. Using the simplified cost analysis, we formulate a novel cost decay function that simultaneously accounts for the commonality among products and the *intended production volume*. A cost objective is subsequently developed on the basis of the cost decay function.

3.1 A Simplified Cost Analysis of a Product Family. Park and Simpson [39] have developed a cost estimation framework for product families. Incorporation of such a cost estimation model into the CP^3 framework is a fruitful topic for future research. In this paper, a simplified cost analysis has been performed to achieve the following specific objectives:

(1) To formulate a cost objective that is more meaningful than (i) a penalty function based on design variable differences or (ii) the inverse of the aggregate commonality.
(2) To investigate the effect of the *intended production volume* of each product variant, on the platform planning process.

The cost analysis is based on the typical product family structure, a representative illustration of which is shown in Fig. 2.

Mathematical details of the cost analysis can be found in the paper by Chowdhury et al. [40]. Chowdhury et al. [40] represents the total cost of the product family as

$$\begin{aligned} &C_F = f_{\rm cost}(X, m, \lambda), \quad \text{where} \\ &m = \left[\, m^1 \quad m^2 \ \cdots \ m^k \ \cdots \ m^N \,\right]^T \end{aligned} \tag{12}$$

In Eq. (12), m^i represents the *intended production volume* for product-i in the family. The vector m can be determined by balancing marketing and manufacturing objectives [5], as well as through demand modeling [41]. Considering the *intended production volume* to be an input parameter for the platform planning process, the cost objective for optimization can be expressed as $f_{\rm cost}(X, \lambda)$.

3.2 Cost Decay Function. In the absence of specific information regarding the cost of manufacturing a family of products, it is challenging to obtain an accurate expression to represent the cost objective function $f_{\rm cost}$. In order to pursue a practical solution approach, the cost objective is simplified as

$$f_{\rm cost}(X, \lambda) = f_{{\rm cost},X}(X) \times f_{{\rm cost},\lambda}(\lambda) \tag{13}$$

where $f_{{\rm cost},X}$ and $f_{{\rm cost},\lambda}$ are independent functions of the design vector X and the commonality matrix λ, respectively. A generic expression for the function $f_{{\rm cost},X}$ is not readily available; the function $f_{{\rm cost},X}$ is assumed to be a constant in this paper. The prospect of obtaining an explicit expression for the function $f_{{\rm cost},\lambda}$ is also restricted in the absence of prior commercial information regarding the product. As a result, a power decay function is assumed to represent the cost; this function is called the cost decay function. The cost decay function simultaneously accounts for (i) the commonality among products and (ii) the *intended production volume* for each kind of product. The cost decay function is defined as

$$\begin{aligned} &CDF_j^k = \left(\frac{1 - c_2}{1 - {c_3}^{-c_1}}\right)\left(\left(M_j^k\right)^{-c_1} - {c_3}^{-c_1}\right) + c_2 \\ &M = \lambda m \end{aligned} \tag{14}$$

where CDF_j^k represents the contribution of the jth variable toward the cost of manufacturing one unit of product-k. The coefficients c_1, c_2, and c_3 control this cost function.

A representative example of the nature of this decay function, on a normalized scale of 0–1, is shown in Fig. 3. This cost function has been developed based on the following hypotheses:

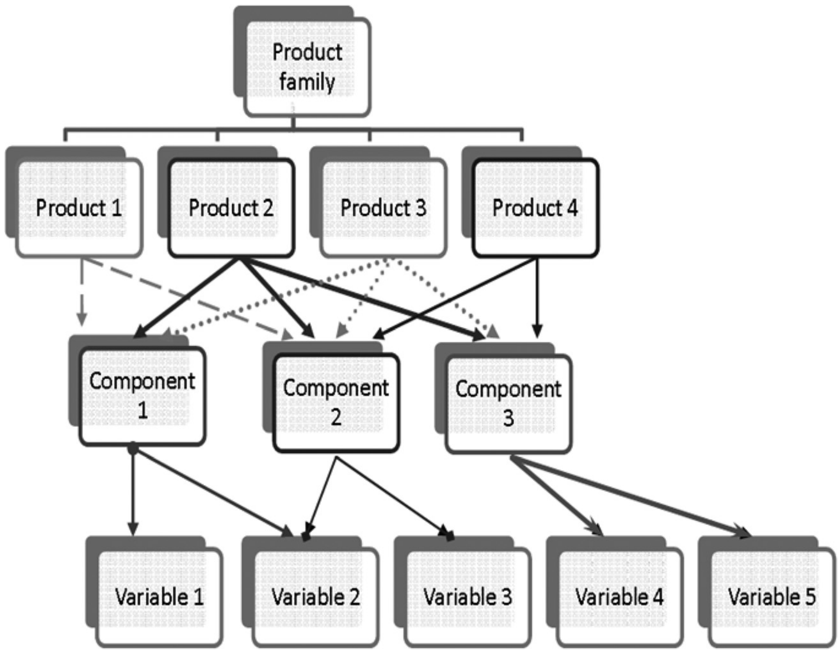

Fig. 2 Typical product family structure

(1) The cost per product would decrease as more products share a particular design variable value.
(2) The marginal cost reduction per product, owing to commonalities, would decrease with increasing production volume.

To provide a better understanding of the CDF, it is important to explain the role of each coefficient:

- The coefficient c_1 controls the rate of decrease of the cost contribution per product. In other words, c_1 is the measure of the sensitivity of "*the marginal cost reduction per product*" to "*the number of products that share a particular attribute*". A higher value of c_1 indicates that *the cost reduction per product* owing to commonalities decreases faster with the volume of production.
- The coefficient c_2 provides a reference lower limit of the cost contribution. In other words, the expression $CDF_j^k - c_2$ represents the per-product overhead cost contribution of the jth attribute. Therefore, if the jth attribute is not shared by more than one product, the per-product overhead cost with respect to the jth attribute will be $1 - c_2$; if the jth attribute is shared by a large number of products, the per-product overhead cost with respect to the jth attribute will become progressively negligible.
- The coefficient c_3 represents the value of the intended production volume beyond which the per-product overhead cost can be considered negligible.

Using the *CDF*, the overall cost of manufacturing the product family is determined by

$$f_{\text{cost}} = f_{\text{cost},X} \times \sum_{k=1}^{N} \left(m^k \sum_{j=1}^{n} \text{CDF}_j^k \right) \tag{15}$$

A function depicting the trend of cost decay (as presented in this paper) is uniquely helpful in investigating potentially critical factors in product platform planning. In the case of commercial product family development, the coefficients c_1, c_2, and c_3 can be derived by solving a standard function estimation problem. However, this cost function is not intended for the comprehensive cost modeling of commercial products. As an actual cost function becomes available, it can be readily used in the approach

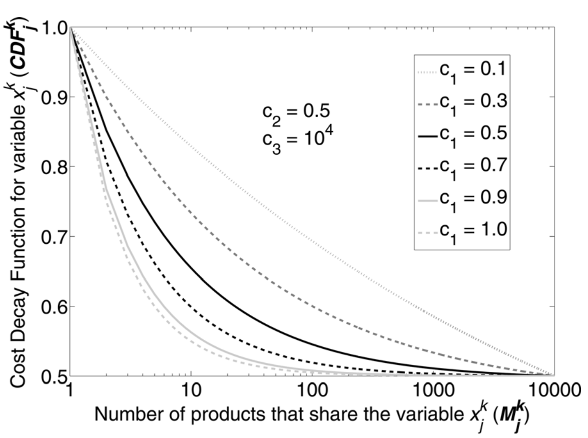

Fig. 3 Cost per product with respect to the total number of products that share a design variable

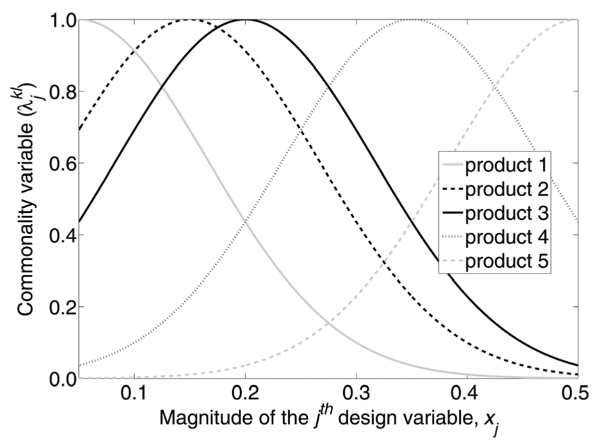

Fig. 4 PSMF for a 5-product family

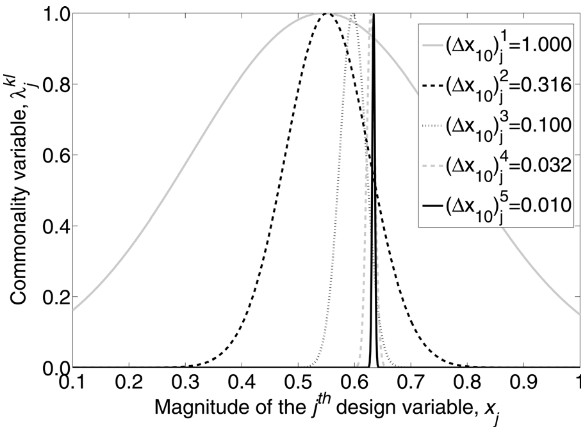

Fig. 5 Five consecutive stages of PSMF application for the commonality between two products (*k* and *l*) with respect to the *j*th design variable

presented in this paper. Accurate response surface based cost functions can also be developed, when adequate data are available regarding (i) the intended volume of production, (ii) the cost of manufacturing due to each attribute, and (iii) the overhead costs.

4 CP^3 Optimization Strategy

In this section, an effective continuous approximation of the CP^3 model is developed. This approximation facilitates the solution of the CP^3 model using standard continuous optimization algorithms. A stepwise pseudocode is also provided to explicate the overall platform planning process developed in this paper. In the later part of this section, we provide a brief description of the particle swarm optimization algorithm that is implemented to optimize the CP^3 model.

4.1 Platform Segregating Mapping Function. The combination of discrete (binary numbers $\{0,1\}$) and continuous design variables in the CP^3 model present a classical mixed integer problem. The high number of binary-integer *commonality variables* λ_j^{kl}, involved in the platform planning model, demands extensive computational resources. A new platform segregating mapping function is proposed to convert the mixed integer problem into a tractable continuous variable problem. Prior to the investigation of this new solution methodology, the commonality constraint (from Eq. (4)) is reformulated as

$$X^T \Lambda X \leq \varepsilon \tag{16}$$

where ε is the aggregate tolerance specified to allow for platform creation. A careful analysis of the modified *commonality constraint* (Eq. (16)) indicates that, for any two products k and l, a *commonality matrix* variable (λ_j^{kl}) is a decreasing function of the square of the corresponding design variable difference ($\Delta x_j^{kl} = |x_j^k - x_j^l|$). At the same time, the design variable differences between various pairs of products are often not independent of each other. A function that can represent the commonality between two products (k and l) with respect to a particular design variable (j) is similar to the function that relates the commonality variable λ_j^{kl} to the design variables x_j^k and x_j^l. This function must have the following properties:

1. The function must be continuous (defined at $\Delta x_j^{kl} = 0$) and well behaved.
2. The function must have a maximum at $\Delta x_j^{kl} = 0$ and must then decrease when Δx_j^{kl} increases (asymptotically tending to zero).
3. These functions, collectively, must allow a coherent consideration of the commonalities between various pairs of products with respect to a particular design variable.

A set of Gaussian kernel functions, collectively called the *platform segregating mapping function*, is developed to approximate the relationship between the commonality variables and the corresponding physical design variables. The required properties listed above are inherent in this set of Gaussian kernel functions. *Interestingly, each Gaussian function in the set also provides a measure of the probability* p_{kl}^j *that the* jth *design variable will be shared between product-k and product-l.* It is helpful to note that other functions that have similar properties can also be implemented to construct the PSMF. The *PSMF* that relates product-l and product-k with respect to the jth design variable is given by

$$\lambda_j^{kl} = a \exp\left(-\frac{\left(x_j^k - x_j^l\right)^2}{2\sigma_j^2}\right) \tag{17}$$

where coefficient a is assumed to be equal to one. In Eq. (17), the design variable values (x_j^l) serve as the mean of the Gaussian kernels and the parameter σ_j represents the standard deviation of the Gaussian kernels for the jth design variable. This standard deviation can be determined by specifying the *full width at one tenth maximum* (Δx_{10}), as expressed by

$$\sigma = \frac{\Delta x_{10}}{2\sqrt{2 \ln 10}} \text{where, } p\left(x_j^l \pm \Delta x_{10}\right) = \frac{1}{10} \tag{18}$$

Table 4 Steps of the PSMF-based CP^3 optimization

Step 1:	Optimize each product using PSO (maximizing performance)
Step 2:	Determine the range for implementing PSO on each x_j
Step 3:	Initiate a random population of size N_{pop}
Step 4:	Set $\overline{\Delta x}_{10} = \overline{\Delta x}_{10}^{max}$ and $i_{stage} = 1$
Step 5:	Simultaneously optimize N products using PSO (solve Eq. 21)
Step 6:	Set $\overline{\Delta x}_{10}^{i_{stage}+1} = \overline{\Delta x}_{10}^{i_{stage}} \times \overline{\Delta x}_{10}^{frac}$, where $\Delta x_{10}^{frac} = \left(\frac{\overline{\Delta x}_{10}^{min}}{\overline{\Delta x}_{10}^{max}}\right)^{\frac{1}{N_{stage}-1}}$
Step 7:	Evaluate $(\Delta x_{10})_j^{i_{stage}+1}$ using Eq. (20)
Step 8:	Choose the optimal configuration as one of the starting points
Step 9:	Initiate a random population of size $N_{pop}-1$, and set $i_{stage} = i_{stage}+1$
Step 10:	If $i_{stage} < N_{stage}$ go to Step 5, else terminate solution

Table 5 Torque requirements of the electric motors

Motor number	1	2	3	4	5	6	7	8	9	10
Torque (N/m)	0.05	0.1	0.125	0.15	0.2	0.25	0.3	0.35	0.4	0.5

where $p(x)$ represents the probability at x. Subsequently, the *commonality matrix* can be expressed as

$$\lambda = PSMF(X) \tag{19}$$

A representative plot of the *PSMF* for a particular design variable (jth variable, normalized), in a sample family of five products, is shown in Fig. 4. In this figure, the design variable for product-k (x_j^k) can be mapped onto the Gaussian kernel of any other product-l (where $l \neq k$), yielding the corresponding commonality variable (λ_j^{kl}). It can be readily observed that, this set of kernel functions (PSMF) provides a unique representation of the product commonalities by converting an essentially combinatorial problem into a tractable continuous problem. This representation can also be used to further investigate the underlying mathematics of product platform planning.

In the CP^3 optimization, initially, an optimal design that maximizes performance is separately obtained for each product variant (in the family). The optimized design variable values, thus determined, are used to set a modified range (Δx_j) for the application of the *PSMF* on each design variable (jth design variable), similar to the approach in the SIO technique [15]. This modified range is used to evaluate the *full width at one tenth maximum* ($(\Delta x_{10})_j$) for the jth design variable, using

$$(\Delta x_{10})_j = \overline{\Delta x}_{10} \times \Delta x_j, \quad \overline{\Delta x}_{10} \in [0, 1] \tag{20}$$

where $\overline{\Delta x}_{10}$ represents the normalized *full width at one tenth maximum*, explicitly specified during the execution of the algorithm.

The CP^3 model is solved using a sequence of N_{stage} particle swarm optimizations, with decreasing values (in a geometric progression) of the parameter $\overline{\Delta x}_{10}$. This multistage optimization results in sharper Gaussian kernels with every subsequent stage, rendering a progressively rigorous application of the *commonality constraint*; this process is illustrated in Fig. 5. This figure shows a five-stage application of the *PSMF*, with the specified initial and final normalized *full width at one tenth maximum* ($\overline{\Delta x}_{10}^{\max} = 1.0$ and $\overline{\Delta x}_{10}^{\min} = 0.1$). In the final stage, design variable values (e.g., x_j^k and x_j^l) residing within the same Gaussian kernel (sharp dome in Fig. 5) would practically indicate that the corresponding products (product-k and product-l) share the jth design variable.

Optimization is performed on the approximated MINLP problem to minimize a simplified weighted-sum of the performance and the cost objectives. The modified optimization problem is defined as

$$\begin{aligned}
&\text{Max } w_1 f_{perf}(X) + (1 - w_1)(-f_{\cos t}(X)) \\
&\quad \text{subject to} \\
&X^T \Lambda X \leq \varepsilon \\
&g_i(X) \leq 0, \quad i = 1, 2, \ldots, p \\
&h_i(X) = 0, \quad i = 1, 2, \ldots, p \\
&\quad \text{where} \\
&\Lambda = f_{con}(\lambda) \\
&\lambda = PSMF(X) \\
&X = \left[x_1^1 x_1^2 \ldots\ x_1^N \ldots\ x_j^1 x_j^2 \ldots\ x_j^N \ldots\ x_n^1 x_n^2 \ldots x_n^N\right]^T \\
&\lambda_j^{lk} \in B : B = \{0, 1\}
\end{aligned} \tag{21}$$

In this problem definition, a value of 0.5 is used for the objective weight w_1, and the term *PSMF* (X) is given by Eq. (17). The objectives f_{perf} and f_{cost} in Eq. (21) are normalized (by dividing by the total number of products manufactured). As is well known, the weighted sum method entails important limitations associated with nonconvex Pareto frontiers. Importantly, we note that the approach presented in this paper can be implemented using any other more powerful method, such as physical programming [42]. The process of application of the *PSMF* technique using particle swarm optimization is summarized by the pseudocode is Table 4.

Neither the optimization of the CP^3 model nor the application of the PSMF-based solution strategy is restricted to the use of the PSO algorithm. The approximated continuous optimization problem provided by the PSMF technique can also be solved using other standard algorithms, which include but are not limited to the real-coded NSGA-II algorithm [36], the Strength Pareto Evolutionary Algorithm [43], the differential evolution algorithm [44], and the modified predator-prey (MPP) algorithm [37]. Population based algorithms are preferable in this case, since the *commonality constraint* is expected to be multimodal. Some of the listed algorithms, such as NSGA-II and MPP, are typically useful for multi-objective optimization; these algorithms can be leveraged to explore a bi-objective optimization scenario, with performance and cost as separate objectives. The original MINLP problem yielded by the CP^3 model can be directly solved using typical MINLP solvers such as Branch and Bound and Cutting Plane techniques, or using binary Genetic Algorithms (e.g., bin-NSGA-II). However, solving the MINLP problem directly may prove to be unreasonably challenging and computationally expensive, owing to the high dimensionality of the commonality variables.

4.2 Constrained Particle Swarm Optimization Algorithm. PSO is one of the most well known stochastic optimization algorithms, initially coined by an electrical engineer (Russel Eberhart) and a social psychologist (James Kennedy) in 1995 [35]. Later, several advanced variations of the algorithm have appeared in the literature, and have been used in popular commercial optimization packages. The PSO algorithm used in this project has been derived from the unconstrained version presented by Colaco et al. [45]. The basic steps of the algorithm are summarized as

Table 6 Design variable limits for the electric motors

Design variable	Lower limit	Upper limit
Number of turns on the armature (N_a)	100	1500
Number of turns on each field pole (N_f)	1	500
Cross-sectional area of the armature wire (A_{wa})	0.01 mm^2	1.00 mm^2
Cross-sectional area of the field pole wire (A_{wf})	0.01 mm^2	1.00 mm^2
Radius of the motor (r_o)	10.00 mm	100.00 mm
Thickness of the stator (t)	0.50 mm	10.00 mm
Stack length of the motor (L)	1.00 mm	100.00 mm
Current drawn by the motor (I)	0.1 A	6.0 A

Table 7 User-defined parameters in the PSMF and the CDF

Parameter	Value
E	1×10^{-06}
C_1	0.2
C_2	0.1
C_3	10,000
$\overline{\Delta x}_{10}^{\max}$	1.00
$\overline{\Delta x}_{10}^{\min}$	0.01
N_{stage}	12

Table 8 User-defined constants in PSO

Constant	Value: step 1	Value: step 5
α	0.5	0.5
β_g	1.4	1.4
β_l	1.4	1.4
Population size (N_{pop})	140	700
Maximum allowed function calls	25,000	40,000

$$
\begin{aligned}
x_i^{t+1} &= x_i^t + v_i^{t+1} \\
v_i^{t+1} &= \alpha v_i^t + \beta_l r_1 (p_i - x_i^t) + \beta_g r_2 (p_g - x_i^t)
\end{aligned}
\tag{22}
$$

where, vector x_i^t represents the ith member of the population (swarm) at the tth iteration, r_1 and r_2 are random numbers between 0 and 1, vector p_i represents the best candidate solution found for the ith member, vector p_g represents the best candidate solution for the entire population, and α, β_l, and β_g are user defined constants.

The constrained non-domination technique [36] is used to compare solutions within a constrained optimization scenario. In this technique, solution-i is said to dominate solution-j if,

(1) solution-i is feasible and solution-j is infeasible or,
(2) both solutions are infeasible and solution-i has a smaller constraint violation than solution-j or,
(3) both solutions are feasible and solution-i weakly dominates solution-j.

If none of the above conditions apply then both of the solutions are considered nondominated with respect to each other.

5 Application of the CP^3 Framework

The product platform planning framework developed in this paper is applied to design a family of universal electric motors. A brief description of the electric motor example is presented in this section, followed by the problem definition for this case study. The results of this product family design case study is then illustrated and discussed.

5.1 Test Problem Description: Universal Electric Motor. Universal electric motors are capable of delivering more torque than any other single phase motors, and can operate using both direct current and alternating current [46]. As a result of such high performance characteristics, universal motors have been used in a variety of applications, e.g., electric drills, electric saws, blenders, vacuum cleaners, and sewing machines [17]. Extensive analysis and detailed equations related to the design of the universal electric motor can be found in the paper by Simpson et al. [47]. The CP^3 framework is applied to develop a scale-based product family of ten universal electric motors that are required to satisfy different torque specifications (T_{rq}), as shown in Table 5. Each motor is also subjected to other design constraints, regarding (i) the output power (P_{out}), (ii) the total mass (M_{total}), (iii) the efficiency (η), (iv) the magnetization intensity (H), and (v) the ratio of the outer radius (r_o) to the thickness (t) of the stator.

The design optimization of the family of universal electric motors involves simultaneous (i) maximization of the efficiency of the motors and (ii) minimization of the cost of the family of motors chiefly attributed to platform planning. The cost minimization demands selection of *platform/sub-platform design variables* and subsequent identification of sub-families within the product family, which is implicit to the MINLP problem presented by the CP^3 model. The design of each motor involves eight design variables; the corresponding variable limits are given in Table 6.

The performance of each individual motor is equivalent to its efficiency. The net performance of the universal motor family, minimized in Step 4 of the pseudocode (Table 4), is formulated to be

$$
f_{\text{perf}} = \frac{1}{10}\sum_{k=1}^{10}\eta^k \tag{23}
$$

where η^k is the efficiency of the motor-k. The cost function component, $f_{\text{cost},X}$ (Eq. 13), is assumed to be a constant equal to one. The cost function for the universal motor family is expressed as

$$
f_{\text{cost}} = \sum_{k=1}^{10}\left(m^k \sum_{j=1}^{7}\mathrm{CDF}_j^k\right) \tag{24}
$$

Thereby, the optimization problem for Step 5 of the pseudocode (Table 4) is given by

$$
\begin{aligned}
&\text{Max } w_1 f_{\text{perf}}(X) + (1-w_1)(-f_{\text{cost}}(X)) \\
&\text{subject to} \\
&\left.\begin{aligned}
&T^k = T^k_{rq}\ \ \forall k=1,2,\ldots,N \\
&P_{\text{out}} = 300\,\mathrm{W}\ \ \forall k=1,2,\ldots,N \\
&M^k_{\text{total}} \le 2\,\mathrm{kg}\ \ \forall k=1,2,\ldots,N \\
&H^k \le 5000\,\mathrm{A.turns/m}\ \ \forall k=1,2,\ldots,N \\
&\eta^k \ge 0.15\ \ \forall k=1,2,\ldots,N \\
&\frac{r_0^k}{t^k} \ge 1\ \ \forall k=1,2,\ldots,N
\end{aligned}\right\}\text{Physical design constraint} \\
&\left.\begin{aligned}
&X^T \Lambda X = 0 \\
&\text{where} \\
&\Lambda = f_{con}(\lambda) \\
&\lambda = PSMF(X)
\end{aligned}\right\}\textit{Commonality constraint} \\
&\text{and where} \\
&X = [N_C\ N_S\ A_{wa}\ A_{wf}\ r_0\ t\ L\ I]^T \\
&\lambda_j^{lk} \in B : B = \{0,1\} \\
&k,l = 1,2,\ldots\ldots,10;\ j=1,2,\ldots\ldots,8;
\end{aligned}
\tag{25}
$$

Table 9 Optimized product platform plan for motor family-1: production volume = 1

Variables	Motor-1	Motor-2	Motor-3	Motor-4	Motor-5	Motor-6	Motor-7	Motor-8	Motor-9	Motor-10
N_a	—	—	—	—	—	—	A	—	A	—
N_f	—	—	—	—	—	—	—	—	—	—
A_{wa}	—	—	B	—	—	—	B	—	—	—
A_{wf}	C	D	C	C	D	C	—	E	E	—
r_o	F	G	F	—	—	—	G	—	—	—
t	H	I	H	I	I	I	I	I	I	—
L	—	J	J	—	—	—	—	J	—	—

Table 10 Optimized product platform plan for motor family-2: production volume = 100

Variables	Motor-1	Motor-2	Motor-3	Motor-4	Motor-5	Motor-6	Motor-7	Motor-8	Motor-9	Motor-10
N_a	—	—	—	—	—	—	A	—	A	—
N_f	—	—	B	—	—	B	—	—	—	—
A_{wa}	C	—	C	-	—	—	—	—	—	—
A_{wf}	D	—	—	D	—	D	—	—	E	E
r_o	—	—	—	—	—	—	—	—	—	—
t	F	G	F	G	—	G	G	G	G	—
L	H	I	H	—	—	—	J	I	J	—

In Eq. (25), the objective functions are normalized, w_1 is equal to 0.5, and T^k is the torque generated by motor-k.

Contrary to the physical components, the variable, current (I), does not participate in platform planning, since: (i) any magnitude of current can be drawn by the motor based on its other characteristics and constraints, and (ii) current does not explicitly contribute to the manufacturing/tooling expenses for the motors [18]. Therefore, current is expressed as a function of other design variables, using the power equality constraint $\left(P^k_{\text{out}} = 300\,\text{W}\right)$. The number of design variables is reduced to seven, and the power equality constraint is relaxed. The inequality constraints representing the upper and lower limits of current are retained.

The CP^3 framework is applied to design four different motor families that are classified on the basis of their *intended production volume*. The *intended production volume* vector for the four motor families are

$$\begin{aligned}
&\text{Motor Family} - 1: m = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]^T\\
&\text{Motor Family} - 2: m = [100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100]^T\\
&\text{Motor Family} - 3: m = [10{,}000 \quad 10{,}000 \quad 10{,}000 \quad 10{,}000 \quad 10{,}000 \quad 10{,}000 \quad 10{,}000 \quad 10{,}000 \quad 10{,}000 \quad 10{,}000]^T\\
&\text{Motor Family} - 4: m = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100]^T
\end{aligned} \tag{26}$$

It is helpful to note that the volume of production varies among the product variants in the fourth product family: 1 unit of each of the first five motors is produced, and 100 units of each of the last 5 motors are produced. The user-defined parameters in the *PSMF* and the *CDF* are given in Table 7.

5.2 CP^3 Application: Results and Discussion. Initially, optimization is performed on each motor individually (using PSO), in order to maximize the efficiency η^k, subjected only to the physical design constraints specified in Eq. 25 (Step 1 of the pseudocode). The user-defined constants in PSO, for this step, are given in Table 8. A new set of variable limits is determined from the highest and lowest values of each optimized design variable, among the ten motors. The new design variable limits are used to execute Step 4—Step 9 of the pseudocode (Table 4). The PSO user-defined constants for optimizing the entire motor family (Step 5 of the pseudocode) are also given in Table 8.

Feasible solutions were successfully obtained for each of the four motor families. The *optimized platform plans* obtained for the four motor families are illustrated in Tables 9–12. Each uppercase alphabet in Table 2, represents a platform; blocks in Table 1, displaying similar alphabets, imply that the corresponding products are members of a particular platform (that share the particular design variable). A block displaying the "–" symbol represents a nonplatform (scaling) design variable value, thereby implying that the corresponding design variable value is not shared by more than one product. It is observed from these four tables that the pattern of platform formation is characteristically different for the four cases. This observation substantiates the sensitivity of the platform planning process to the *intended production volume*, a factor that is largely overlooked in product family design methodologies.

Ten sub-platforms are formed in motor families-1 and -2, nine sub-platforms are formed in motor family-3, and only five platforms are formed in motor family-4. Expectedly, the overall membership of the sub-platforms (number of products sharing a variable) for motor family-1 is higher than that for the family-2 and family-3. The "*all-common or all-distinct*" approaches do not allow the formation of such sub-families (as given by the CP^3 framework), which restricts their applicability to a wider variety of commercial product families. An "*all-common or all-distinct*" approach would also demand a higher compromise of the product performances to achieve commonality among products [17]. *It is helpful to note that a product platform plan yielded by an "all-common or all-distinct" approach would be a subset/specific-case of the generalized scenario allowed by the CP^3 model.*

The overall performance and cost objectives are complex nonlinear functions of the product attributes, which makes it challenging to conclusively investigate why certain attributes are shared and certain others are not in the optimized platform plan. Nevertheless, some interesting observations can be made from the

Table 11 Optimized product platform plan for motor family-3: production volume = 10,000

Variables	Motor-1	Motor-2	Motor-3	Motor-4	Motor-5	Motor-6	Motor-7	Motor-8	Motor-9	Motor-10
N_a	—	—	—	—	—	—	A	—	A	—
N_f	—	—	—	—	—	—	—	—	—	—
A_{wa}	—	—	—	—	—	B	B	—	—	—
A_{wf}	C	—	C	C	—	C	—	—	—	—
r_o	D	E	—	—	—	—	E	D	—	—
t	—	F	—	—	—	F	F	F	F	—
L	G	H	G	—	—	—	I	H	I	—

Table 12 Optimized product platform plan for motor family-4: production volume = 1 (first five motors), 100 (last five motors)

Variable	Motor-1	Motor-2	Motor-3	Motor-4	Motor-5	Motor-6	Motor-7	Motor-8	Motor-9	Motor-10
N_a	—	—	—	—	—	A	—	—	—	A
N_f	—	—	—	—	B	—	B	—	—	—
A_{wa}	—	—	C	—	—	—	—	—	—	C
A_{wf}	—	—	—	—	—	—	—	—	—	—
r_o	—	—	—	—	—	—	—	—	—	—
t	—	—	—	D	—	D	D	D	D	—
L	—	—	—	—	E	—	—	—	E	—

resulting platform plans. For example, motor-10 and motor-5 are found to be the least likely to share attributes with the other motors. We also observe that, the differing platform combinations for the four motor families provide a consistency in performance, whereas their degrees of attribute sharing are different. This trend can be attributed to the use of the simple linear aggregate of objectives. A more comprehensive multiobjective solution approach should provide further insights into the trade-offs between performance and costs, and how these trade-offs are influenced by the volume of production.

Another interesting observation is the formation of a significantly lower number of sub-platforms in the case of family-4. This observation can be attributed to the relatively biased consideration of the performance and the cost objectives in the case of family-4. The overall performance objective is determined by the aggregate of the performances of the individual product variants, irrespective of their volumes of production; hence, equal importance is given to the performance of each product variant. On the other hand, a careful analyses of Eq. (15) and the cost decay function show that the volumes of production of each product variant is incorporated within the cost objective. As a result, product variants with lower volumes of production (e.g., the first five motors) have a significantly lower impact on the cost (or commonality) aspect of the product platform planning process. Out of a total production of 505 units in the family, only 1 unit of each of the first five motors is present; the tendency of platform formation among the first five motors is therefore marginal. Unlike the cost function proposed in this paper, when the product overhead cost for each product variant is assumed to be completely independent of the corresponding volume of production, the overhead cost of each variant would be directly additive. In that case, family-4 is expected to illustrate a similar degree of commonality as that of the other three families.

In order to explain the tendency of platform formation (for a particular product family), a parameter called the *extent of commonality* (E_C) is defined. This parameter is given by the ratio between the number of off-diagonal "ones" (obtained through CP^3), and the total number of off-diagonal elements in the *commonality matrix*, expressed as

$$E_C = \frac{1}{nN(N-1)} \sum_{j=1}^{n} \sum_{k \neq l} \lambda_j^{kl}, \forall k = 1, 2, \ldots, N \text{and} l = 1, 2, \ldots, N \tag{27}$$

The (E_C) is somewhat similar to the commonality objective used in the product family optimization technique presented by Khajavirad et al. [18]. Figure 6 illustrates the *extent of commonality* for the first three families with respect to the *intended production volume*.

The tendency of platform formation is observed to decrease with increasing capacity of production (from Fig. 6). This seemingly counterintuitive observation can be attributed to the nature of cost variation provided by the cost decay function, presented in this paper. The latter is based on the following likely commercial scenario—"The marginal reduction in cost per product, owing to commonalities among products, decreases with increasing production volume of each individual kind of products in the family." It is also found that the extent of commonality is lowest in family-4 (approximately 0.04); this observation can be attributed to the negligible impact of the first five motors, which have a lower production volume.

The proposed variation of "the cost per product" with "the number of products that share a variable" presents a decreasing gradient. For different commercial products, the actual variation of cost with the *production volume* and the *commonalities among products* may follow a trend that is different from this artificial cost decay function. For example: (i) if the variation of "cost per product" with "the number of products that share a variable" becomes linear, platform planning will become practically insensitive to the intended production volume; and (ii) if the variation presents an increasing gradient, platform planning will seek to enhance the commonalities among products, with increasing production volume. Hence, the authors do not claim uniqueness regarding the variation of the *extent of commonality* (achieved through product family optimization) with the *intended production volume*. The primary objective of this investigation is to illustrate how the *intended production volume* is a crucial factor in product family optimization.

Figures 7(*a*) and 7(*b*) show the mean and the standard deviation of the efficiencies of the ten motors for each family. The mean (approximately 0.82) and the standard deviation (approximately 0.083) of the motor efficiencies remain approximately the same with increasing *intended production volume*, while the *extent of commonality* changes (as seen from Figs. 6, 7(*a*), and 7(*b*)). In order to preserve the consistency in performance and cost (subject to the defined cost decay function), the platform plan is expected to shift toward lower commonality when the *intended production volume* is increased (as illustrated by Figs. 6, 7(*a*), and 7(*b*)).

Product family optimization involves maximization of the net performance of the different products, which is generally an aggregation of the individual product performances. This approach may lead to an unfair distribution of product performances within the family. In this context, a more flexible approach

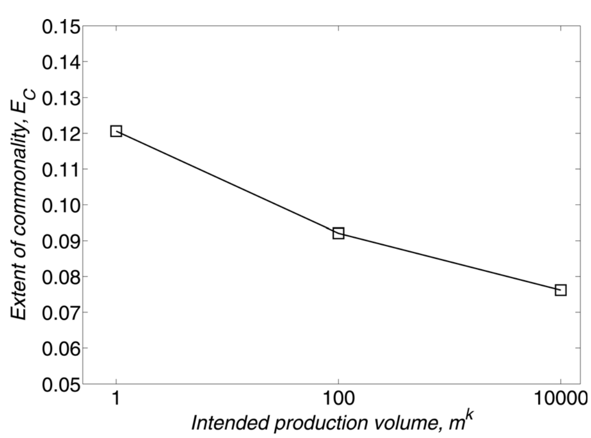

Fig. 6 Variation of the *extent of commonality with the intended production volume*

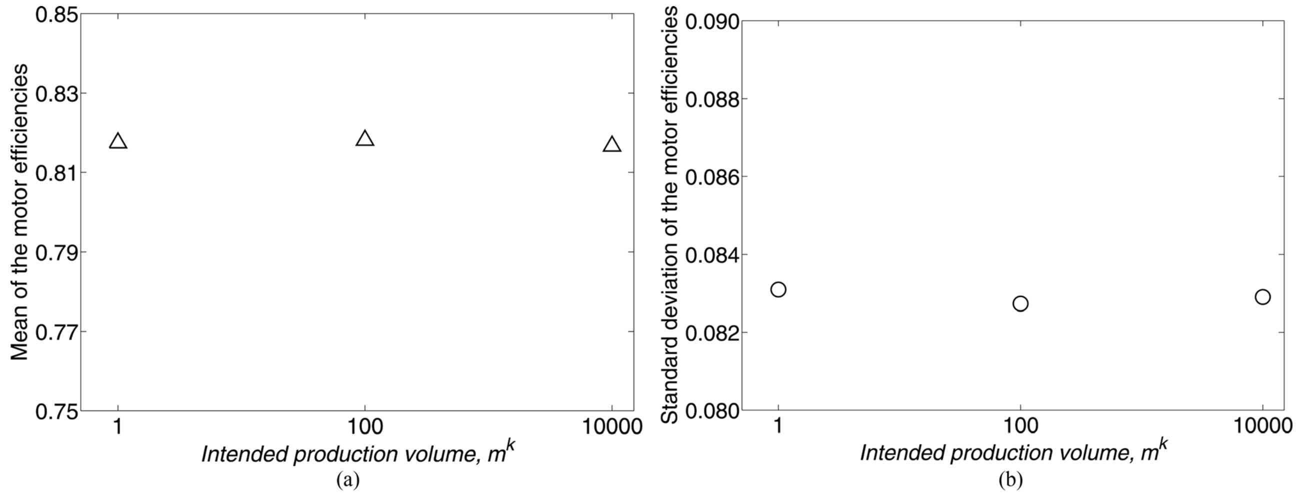

Fig. 7 (*a*) The mean and (*b*) the standard deviation of the efficiencies of the ten motors for each family

should allow for the explicit consideration of the standard deviation in the performances of the different products in a family—for example, by setting a maximum allowed value of the standard deviation as a design constraint. In addition, market factors often introduce specific requirements regarding the performance of individual products.

6 Concluding Remarks

The (CP^3) framework lays the foundation of a unified approach that seeks to capture the full potential of the product family paradigm. The CP^3 technique introduces an encompassing mathematical model of the platform planning process (CP^3 model), which yields a mixed integer nonlinear programming problem. A *commonality matrix* is defined to classify (i) the variables that are shared by certain products and (ii) the variables that are not shared by more than one product. This *commonality matrix* also provides welcome flexibility to include or exclude a design variable (in a product), thereby accounting for the instantiation level of modular product family design. Another important contribution of this formulation is the scope of designing sub-families of products—where design variables are shared by a subset of products in the family. The *commonality constraint* developed using the *commonality matrix* provides a robust framework, where the simultaneous (i) selection of platform/scaling variables, and (ii) optimization of the design variable values can be readily performed. In addition to these expedient features, the resulting platform planning model (CP^3 model) is independent of the optimization strategy. To the best of author's knowledge, such a generalized mathematical formulation of the product platform planning process has favorably unique features. Nevertheless, the commonality matrix definition involves a certain degree of redundancy: e.g., the knowledge of the commonalities between product-1 and product-2 and that between product-2 and product-3 can provide substantial (not necessarily complete) information regarding the commonality between product-1 and product-3. The assumption that the commonality variables are independent in the current CP^3 model does not capture this flow of information. Future advancement of the CP^3 model should address this redundancy and help reduce the model-dimensionality and the associated computational expense.

An efficient conversion of the MINLP problem into a continuous problem is expected to reduce the net computational expense. To this end, the novel platform segregating mapping function (PSMF) is developed and applied. A Cost Decay Function (CDF) is introduced to represent the variation of the cost of the product family with the *intended production volume* and the designed commonalities among products. This CP^3 framework is applied using particle swarm optimization to design a family of ten universal electric motors. It is observed that the set of variable based platforms, obtained through CP^3, is distinct for different *intended production volumes*. Credible commercial information appears to be necessary to further investigate this scenario.

In this paper, a linear aggregate function has been used to combine the performance and the cost objectives. For future work, a more powerful multi-objective setting should be investigated to explore the trade-offs between individual product performances and the net cost benefit of platform planning. Further advancement of the CP^3 framework to allow a more comprehensive consideration of the product architecture and the availability of module options is also a fruitful topic for future research. Subsequent exploration of module-based product family applications would more clearly establish the true potential of this new method.

Acknowledgment

Support from the National Science Foundation Awards CMMI 0946765 and CMMI 1100948 is gratefully acknowledged.

References

[1] Meyer, M. H., and Lopez, L., 1995, "Technology Strategy in a Software Products Company," J. Prod. Innovation Manage., **12**(4), pp. 294–306.
[2] Meyer, M. H., and Lehnerd, A. P., *The Power of Product Platforms: Building Value and Cost Leadership* (The Free Press, New York, 1997).
[3] Robertson, D., and Ulrich, K., 1998. "Planning Product Platforms," Sloan Manage. Rev., **39**(4), pp. 19–31.
[4] Simpson, T. W., Siddique, Z., and Jiao, R. J., *Product Platform and Product Family Design: Methods and Applications* (Springer, New York, 2006).
[5] 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.
[6] Simpson, T. W., Chen, W., Allen, J. K., and Mistree, F., 1996, "Conceptual Design of a Family of Products Through the Use of the Robust Concept Exploration Method," In 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Report No. AIAA-96-4161-CP, AIAA, pp. 1535–1545.
[7] Simpson, T. W., 2004, "Product Platform Design and Customization: Status and Promise," Artif. Intell. Eng. Des. Anal. Manuf., **18**(1), pp. 3–20.
[8] Uzumeri, M., and Sanderson, S. W., 1995, "A Framework for Model and Product Family Competition," Res. Policy, **24**(4), pp. 583–607.
[9] Sanderson, S. W., and Uzumeri, M., 1997. *Managing Product Families* IRWIN, USA.
[10] Sabbagh, K., *Twenty-First Century Jet: The Making and Marketing of Boeing 777* (Scribner, 1996).
[11] Kobe, G., 1997, "Platforms—Gm's Seven Platform Global Strategy," *Automot. Ind.*, **177**(50).
[12] Dai, Z., and Scott, M. J., 2006, "Effective Product Family Design Using Preference Aggregation," ASME J. Mech. Des., **128**(4), pp. 659–667.
[13] Simpson, T. W., 2006, "Product Platform and Product Family Design: Methods and Applications," Springer, New York, ch. *Methods for Optimizing Product Platforms and Product Families: Overview and Classification*, pp. 133–156
[14] Chowdhury, S., Messac, A., and Khire, R., 2010, "Developing a Non-Gradient Based Mixed-Discrete Optimization Approach for Comprehensive Product

Platform Planning (CP^3)," 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, AIAA.
[15] 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, Report no. AIAA 2006-6924, AIAA.
[16] Khire, R., and Messac, A., 2008, "Selection-Integrated Optimization (sio) Methodology for Optimal Design of Adaptive Systems," ASME J. Mech. Des., **130**(10), 101401.
[17] Khire, R. A., 2006, "Selection-Integrated Optimization (sio) Methodology for Adaptive Systems and Product Family Optimization," Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY.
[18] Khajavirad, A., Michalek, J. J., and Simpson, T. W., 2009, "An Efficient Decomposed Multiobjective Genetic Algorithm for Solving the Joint Product Platform Selection and Product Family Design Problem With Generalized Commonality," Struct. Multidiscip. Optim., **39**(2), pp. 187–201.
[19] Chen, C., and Wang, L. A., 2008, "Modified Genetic Algorithm for Product Family Optimization With Platform Specified by Information Theoretical Approach," J. Shanghai Jiaotong Univ., **13**(3), pp. 304–311.
[20] Messac, A., Martinez, M. P., and Simpson, T. W., 2002, "Effective Product Family Design Using Physical Programming," Eng. Optimiz., **124**(3), pp. 245–261.
[21] Messac, A., Martinez, M. P., and Simpson, T. W., 2002, "Introduction of a Product Family Penalty Function Using Physical Programming," ASME J. Mech. Des., **124**(2), pp. 164–172.
[22] Rangavajhala, S., Khire, R., and Messac, A., 2007, "Decision Making in Product Family Optimization Under Uncertainty," 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Report No. AIAA 2007-1894, AIAA.
[23] Rai, R., and Allada, V., 2007, "Modular Product Family Design: An Agent Based Optimization Technique," Int. J. Prod. Res., **41**(17), pp. 4075–4098.
[24] Fujita, K., and Yoshida, H., 2001, "Product Variety Optimization: Simultaneous Optimization of Module Combination and Module Attributes," ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Report No. DETC2001/DAC-21058.
[25] Fujita, K., and Yoshida, H., 2004, "Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes," Concurr. Eng., **12**(2), pp. 105–118.
[26] Stone, R. B., Wood, K. L., and Crawford, R. H., 2000, "A Heuristic Method to Identify Modules From a Functional Description of a Product," Des. Stud., **21**(1), pp. 5–31.
[27] Dahmus, J. B., Gonzalez-Zugasti, J. P., and Otto, K. N., 2001, "Modular Product Architecture," Des. Stud., 22, pp. 409–424.
[28] Guo, F., and Gershenson, J. K., 2003, "Comparison of Modular Measurement Methods Based on Consistency Analysis and Sensitivity Analysis," ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Report No. DETC2003/DTM-48634.
[29] Jose, A., and Tollenaere, M., 2005, "Modular and Platform Methods for Product Family Design: Literature Analysis," J. Intell. Manuf., 16, pp. 371–390.
[30] Kalligeros, K., de Weck, O., and de Neufville, R., 2006, "Platform Identification Using Design Structure Matrices," Sixteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE), INCOSE.
[31] Sharon, A., Dori, D., and de Weck, O., 2009, "Model-Based Design Structure Matrix: Deriving a dsm From an Object-Process Model," *Second International Symposium on Engineering Systems* CESUN and MIT ESD
[32] Yu, T. L., Yassine, A. A., and Goldberg, D. E., 2007, "An Information Theoretic Method for Developing Modular Architectures Using Genetic Algorithms," Res. Eng. Des., 18, pp. 91–109.
[33] GamsWorld 2010, "MINLP World," http://www.gamsworld.org/minlp/index.htm.
[34] CMU and IBM, 2010, "CMU-IBM Cyber-Infrastructure for MINLP," http://www.minlp.org/index.php.
[35] Kennedy, J., and Eberhart, R. C., 1995, "Particle Swarm Optimization," IEEE International Conference on Neural Networks, no. IV, IEEE, pp. 1942–1948.
[36] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002, "A Fast and Elitist Multi-Objective Genetic Algorithm: Nsga-ii," IEEE Trans. Evol. Comput., **6**(2), pp. 182–197.
[37] Chowdhury, S., and Dulikravich, G. S., 2010, "Improvements to Single-Objective Constrained Predator-Prey Evolutionary Optimization Algorithm," Struct. Multidiscip. Optim., **41**(4), pp. 541–554.
[38] Khajavirad, A., and Michalek, J. J., 2008, "A Decomposed Gradient-Based Approach for Generalized Platform Selection and Variant Design in Product Family Optimization," ASME J. Mech. Des. **130**, 071101.
[39] Park, J., and Simpson, T. W., 2005, "Development of a Production Cost Estimation Framework to Support Product Family Design," Int. J. Prod. Res., **43**(4), pp. 731–772.
[40] Chowdhury, S., Messac, A., and Khire, R., 2010, "Comprehensive Product Platform Planning (cp3) Framework: Presenting a Generalized Product Family Model," 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA, Report No. AIAA 2010-2837.
[41] de Weck, O., Suh, E., and Chang, D., 2003, "Product Family and Platform Portfolio Optimization," ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, ASME, Report No. DETC03/DAC-48721.
[42] Messac, A., 1996, "Physical Programming: Effecrtive Optimization for Computational Design," AIAA J., **34**(1), pp. 149–158.
[43] Zitzler, E., Laumanns, M., and Bleuler, S., 2004, "Metaheuristics for Multiobjective Optimisation," *A Tutorial on Evolutionary Multiobjective Optimization* Springer, New York, pp. 3–37.
[44] Price, K., Storn, R. M., and Lampinen, J. A., 2005, *Differential Evolution: A Practical Approach to Global Optimization* 1st ed., Springer, New York.
[45] Colaco, M. J., Orlande, H. R. B., and Dulikravich, G. S., 2006, "Inverse and Optimization Problems in Heat Transfer," Journal of the Brazilian J. Braz. Soc. Mech. Sci. Eng., **28**(1).
[46] Chapman, S. J., 2002, *Electric Machinery and Power System Fundamentals*, McGraw-Hill, New York.
[47] Simpson, T. W., Maier, J. R. A., and Mistree, F., 2001, "Product Platform Design: Method and Application," Res. Eng. Des., **13**(1), pp. 2–22.