A Method to Improve Platform Leveraging in a Market Segmentation Grid for an Existing Product Line

Ronald S. Farrell
Graduate Research Assistant
Mem. ASME

Timothy W. Simpson
Professor
Mem. ASME

Department of Mechanical
and Nuclear Engineering,
The Pennsylvania State University,
University Park, PA 16802

This paper describes a method for improving commonality in a highly customized low volume product line whose members were originally developed one at a time to meet specific customer requirements. Rather than focusing on redesign of the entire product line, which can often be cost prohibitive, the method is part of a strategy to redesign a limited set of component parts that have the highest potential for cost savings. The method involves a four-step methodology: (1) determine an optimal component solution for each member artifact of an existing market segment grid, (2) test the feasibility of using each optimal component as a platform for the other artifacts, (3) formulate an optimization problem around the feasibility statistics whose solution is a product platform portfolio, and (4) solve the optimization problem for the platform portfolio that can span the existing market segment grid most cost effectively. The proposed method is applied to an example involving the redesign of actuator mounting yokes for an existing set of valves that are used in nuclear power plants. The methodology shows promise for determining a product platform mix that maximizes cost effectiveness yet meets performance requirements. [DOI: 10.1115/1.2829889]

Keywords: product platform, segmentation, commonality, redesign product variety, market

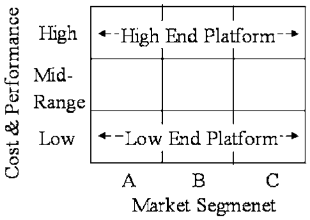
1 Introduction

Commonality is recognized as an effective way to achieve economies of scale and scope, and product families have been successfully employed by various companies such as Sony [1], Black & Decker [2], Lutron [3], and Airbus [4] to address the challenge of providing variety for the marketplace while maintaining commonality between products. A product family is a group of related products that share common features, components, and subsystems, yet satisfy a variety of market niches. The set of common parameters, features, or components that remain constant from product to product within a given product family is referred to as the product platform. The product platform provides the basis for the product family, which is derived through the addition, substitution, or exclusion of one or more modules from the platform [5–9] or by scaling the platform in one or more dimensions [10–14]. In product platform design, it is common to map overall design requirements into an appropriate *market segmentation grid* [2], as shown in Fig. 1. The grid allows for identification of potential leveraging opportunities for the product platform to satisfy effectively a variety of market segments. It is traditional to employ horizontal, vertical, and beachhead approaches as illustrated in Fig. 1 to enable effective platform leveraging both within and across different market segments.

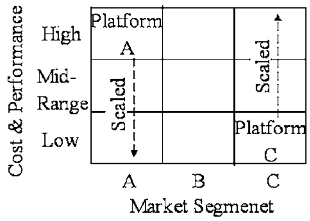
A company's product line could be based totally or in part on a *product platform portfolio*, which is a collection of product platforms that covers the entire market segment grid. A design process centered on a product platform portfolio could result in a cost-effective product development system that can provide the variety that the market demands. An agile manufacturing system combined with a well-designed portfolio can efficiently and proactively change to meet future demand.

In most research, the product platform portfolio and the design and platform variables are established prior to detailed design; however, there has been research involving optimization prior to detailed platform design [15]. Some research endeavors to distinguish between design variables, which change with each family member, and platform variables, which are constant within the family [16–18]. For instance, D'Souza and Simpson [16] employ a nondominated sorting genetic algorithm (GA) to optimize the balance between commonality and performance. Effectively, this optimizes the extent to which the design variables cover the targeted market segments. de Weck, et al. [19] present a method to determine the optimal number of product platforms to maximize overall product family profit with simplifying assumptions. The method is demonstrated for a hypothetical automotive vehicle line that is required to fill seven market segments. The portfolio can vary from one to seven platforms: The seven-platform case corresponds to no leveraging while the single platform case corresponds to the maximum leveraging possible. The method computes the profit for each portfolio from the set of seven, and chooses the one that yields the highest profit as the optimal one; a three-platform portfolio is determined to be optimal for their example. Fujita and Yoshida [20] advocate the simultaneous design of multiple products. Assuming a modular architecture, they propose a simultaneous optimization method for both module combination and module attributes of multiple products. The method considers cost, profit, commonality, and similarity, and hybridizes a GA and a mixed-integer programming method with a branch-and-bound technique, and a constrained nonlinear programming method. This is an extension of other work [21] where optimization of module combination and module attributes is treated separately. In other research, product family selection is optimized based on a performance loss function [22], or optimization is

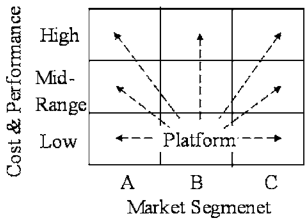
Submitted to ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received May 23, 2006; final manuscript received August 18, 2007; published online February 5, 2008. Review conducted by Kemper Lewis. Paper presented at the ASME 2006 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2006), Philadelphia, PA, September 10–13, 2006.



(*a*) **Horizontal Leveraging**



(*b*) **Vertical Leveraging**



(*c*) **Beachhead Approach**

Fig. 1 Market segmentation grid and platform leveraging strategies [2]

based on combining business and engineering decisions [23]. In a paper involving modular architecture artifacts, a GA-based method is used to optimize module sharing and creation of new modules [23]. In the work by Hernandez et al. [24], the goal is to minimize the impact of commonality on performance using the concept of space of customization.

Development of the method proposed in this paper is motivated by the need to improve commonality in a highly customized low volume product line whose members were originally developed one at a time to meet specific customer requirements, as it can be difficult to achieve and maintain commonality. When a product is unique, it results in high development and production costs that are difficult to predict, and in long and uncertain production times. A manufacturer of these products may eventually develop a quasistandard product line, but since the line is designed one custom product at a time, the full spectrum of product offerings is rarely reviewed to ensure that it is optimal for the business [25]. Focusing on custom products can result in "a failure to embrace commonality, compatibility, or standardization" [26], leading to a proliferation of products and parts with increasing costs and overhead. The failure potential increases by degree for highly customized product lines and is even greater for small firms.

Redesign of a "one-at-a-time" product line can be cost prohibitive, especially for a small firm. However, what may be justifiable is a strategic redesign of a limited set of component parts that have the highest potential for cost saving. A component part redesign effort can employ the product platform approach, and when applied across the market segment, a *component platform portfolio* results.

This paper presents a method for optimizing a component platform portfolio so that market segment grid platform leveraging is maximized. It assumes a product line already exists that fits into an existing market segment grid, and the objective is to redesign individual components that are common to all members of the product line but lack commonality between the members. The method does not rely on the traditional leveraging strategies (horizontal, vertical, and beachhead), as no constraints are imposed on the leveraging strategy. The goal is to minimize manufacturing cost by minimizing the number of component platforms required to span the market segment grid without sacrificing product performance or customer perceived variety, but this can be challenging because it involves a trade-off between minimum cost and maximum performance. An important precursor is a design method that can determine component design parameters that optimize performance for a single artifact from the market segment grid. The proposed method is described in detail in Sec. 2 as a four-step process. In Sec. 3, an example involving the design of yokes on nuclear grade valves is presented that follows the four steps. Section 4 gives closing remarks and discusses the example problem results and potential future work.

2 Proposed Method

The preexistence of design artifacts that make up a market segment grid is assumed. The artifacts should have a common component, each of unique design that is to be designed or redesigned with the goal of maximizing commonality throughout the market segment grid. Improved commonality is achieved through the design of component platforms, where each platform forms the basis for common component design for a group of artifacts. Then, several platforms are required to address all components in the grid, and this collection of platforms is defined as the component platform portfolio. The ultimate goal is to determine the component platform portfolio that optimizes the commonality and performance of the family in the market segmentation grid.

As a precursor, we assign an ordinal (i) to each artifact in the market segment grid. Then, the grid is symbolized by an array (S) with n members, and the elements of S contain sequential numbers from 1 to i to n. At a minimum, a design strategy must exist that can be implemented on artifact i to determine optimal critical component design parameters (X_i^*), which yield an optimal objective function (F_i^*). Description of the four steps in the proposed method follows.

Step 1: Determine an optimal component solution for each member (i) in the market segment grid. Each solution consists of an optimal objective function (F_i^*), an array of optimal design variables (X_i^*), an array of design value constraints associated with performance (P_i), and an array of constraints (B_i) associated with the upper and lower bounds on X_i. The X_i^* array and the means to reformulate the constraint equations B_i and P_i are saved for subsequent steps. The method assumes that the optimal product platform portfolio consists of a subset of the resulting optimal designs. This step mimics the two-step approach first advocated by Nelson et al. [27] and then refined by Fellini et al. [14,28] who first optimize the individual products—to determine what the best possible performance is for each product when there is no commonality—before optimizing the family of products (i.e., their commonality), which is performed in Steps 3 and 4 of our method.

It is not necessary that the resulting designs are globally optimal, as this can be difficult to achieve and prove in general practice because of problem complexity such as ill-behaved functionality or quickly changing market needs. Rather, what is required are feasible designs along with a method for assessing feasibility of design variable constraints and performance constraints. For instance, it is acceptable to start with existing artifact component designs as long as the method exists or is developed and the existing designs are feasible. Generally, the goal of any engineering design problem is to obtain and employ the best available solution, and in this paper, the term "optimal design" is considered synonymous with "the best available solution."

Step 2: For each optimal component, test the feasibility of using it as a platform on each market segment grid member. Assuming that the bounds and performance constraints are in standard form, i.e., the value of the constraint function is less than or equal to zero when the constraint is satisfied, an optimal component (component j) is a *candidate component platform* for a market segment member (member i) when Eq. (1) is satisfied.

$$X_j^* \in \{X_i | B_i(X_i) < 0 \quad \text{and} \quad P_i(X_i) < 0\} \tag{1}$$

Given the n market segment members and the corresponding n optimal components, (n^2-n) tests are required, where $n-1$ component feasibility tests are associated with each of the n market segment members. Notice that testing a component against its own source member is not required as its feasibility is assured in advance. Then, any one test involves inserting each component's optimal solution design variables X_j^* into each market segment member i's constraint equations B_i and P_i, and assessing whether the constraints are satisfied. That is, Step 2 involves testing whether X_j^* is a feasible solution to member i's optimization problem for all j not equal to i.

The best approach that results in efficient feasibility testing is to examine each member (i) in turn so that each optimization problem from Step 1 is reformulated just once. In order to further save computing effort, we test each component (j) in two phases. In Phase 1, we test only the bound constraints (B_i), and in Phase 2, we test the remaining constraints (P_i). The two-phase testing procedure can save effort because bound constraint equations are typically simpler than performance constraint equations, which may involve complex analyses. For instance, performance constraint evaluation may require solving a complex finite element model to determine some physical parameter such as a stress level, a flow rate, or a frequency response to name just a few. It is likely that the required number of more expensive tests (Phase 2) is only a fraction of the required number of Phase 1 tests, which is $n-1$. Then, despite the (n^2-n) required tests, the method has potential for successful application on problems with a large number of artifacts. Similar two-phase approaches are used in other disciplines when the computational expense of some analyses is high. In the aerospace community, for instance, lower fidelity models that are inexpensive to compute are first used to reduce the design space or identify a "reasonable" design space before invoking higher fidelity models that are much more expensive to compute to perform analyses in that reduced region [29,30].

Step 2, Phase 1: Bound Feasibility Test. In Phase 1, the bound constraints are tested for feasibility without regard for performance feasibility, and this postpones reformulation of the complete optimization problem until needed during the Phase 2. For each artifact, $n-1$ bound tests are required, and what results is a collection of bound-feasible candidate component platforms. Let the resulting number of bound-feasible candidates equal n_B, then this number can range from 1 (i.e., only the artifact's own platform is feasible) to n (i.e., all candidate platforms are bound-feasible), but it is probable that the resulting number is significantly less than n.

Step 2, Phase 2: Performance Feasibility Test. In Phase 2, only the bound-feasible candidate component platforms from Step 1 are tested for performance constraint feasibility. Although evaluation of performance constraints may involve costly computations, the effort is reduced since Phase 1 testing has eliminated bound-infeasible candidates. Performance constraint equations need to be formulated only once for each artifact, and each bound-feasible candidate test is a load case that is solved for performance feasibility. What results is a further reduced set of candidates, let it equal n_P, that are both bound feasible and performance feasible, and n_P can range from 1 (i.e., only the artifact's own platform is feasible) to n_B (i.e., all bound-feasible platforms are also performance feasible). Notice that if n_P equals 1, any product platform portfolio must include this artifact's component platform.

Step 3: Formulate an optimization problem to identify the component platform portfolio. From the previous step's result, we construct arrays of candidate platforms (C_j) for each optimal component (j), then each C_j contains the grid member ordinals (i) for which the component (j) satisfies constraints. Then, the component portfolio platform design variables (X^P) consist of n elements, one for each component (j). The value of a portfolio platform design variable (X_j^P) equals the index into the component's C_j array. Each element of X^P is bounded from one to the size of C_j. The design variables define which component is used as a platform for each grid member such that the n used platforms are given by $C(X^P)$.

The goal in the optimization process is to minimize the cost of implementing the resulting component platform portfolio without jeopardizing its ability to meet performance objectives. In general then, achieving this goal involves a trade-off between cost and performance, and the method assumes that a single trade-off metric (T) can be formulated that adequately measures this objective. Given the assumption from Step 1 that the component platform portfolio consists of a subset of the optimal artifact component designs, T can be expressed as a function of the used component platforms given by $C(X^p)$. The generalized optimization problem is thus stated as follows:

$$\begin{gathered}\text{Minimize } T(X^P)\\ \text{Subject to the upper and lower bounds on } X^P\end{gathered} \tag{2}$$

The majority of product platform design methods assume that maximizing commonality is a reasonable surrogate for minimizing cost [31]. If this assumption is employed, then any of the commonality indices available in the literature (see Ref. [32] for a recent review) can be used for the trade-off metric (T) when a more sophisticated metric or cost model is not available. In fact, Khajavirad and Michalek [33] recently argued that the commonality index (CI) introduced by Martin and Ishii [26] captures the tooling cost savings of component commonality better than other commonality metric. If we use this index as our starting point, then an effective trade-off metric (T) for our problem is N—where N is the number of unique ordinals in $C(X^p)$—since we are dealing with a component platform (i.e., a single component that is standardized across as many market segments as possible). For instance, if $C(X^p)$ equals {1, 2, 1, 3, 2}, then N equals 3, and our objective is to thus minimize N in order to maximize commonality and reduce the number of unique component instances needed to span the market segmentation grid. Future work should verify this assumption about maximizing commonality when a suitable cost model is available, and we are aware of no such studies in the literature that have made this comparison.

Step 4: Solve the optimization problem. Finding a solution requires a zero-order algorithm such as the simulated annealing (SA) algorithm or the GA, which is capable of addressing integer design variables. Given the optimal solution (T^*), the components to use as platforms are given by the unique ordinals in $C(X^{p^*})$, and the platforms to use for each grid member is given by $C(X^{p^*})$.

The solution to Eq. (2), which is denoted as T^*, can have two extremes depending on the trade-off metric employed. If T is taken as CI as discussed in the previous step, then the solution, N^*, denotes the number of component platforms needed to span the market segmentation grid. The other extreme occurs when T is so biased toward performance constraints that no commonality is achievable. In this case, the resulting component platform portfolio is defined by $C(X^{p^*})=i$ for every artifact i, which is referred to as the *null platform* [27] in that there is no suitable level of commonality within the product family for the market.

3 Example Problem

Implementation of the four-step process is demonstrated using an example involving the design of yokes in a market segment of nuclear grade valves. Before demonstrating the portfolio optimization process in Sec. 3.2, the fundamentals of valve operation and valve design are discussed.

3.1 Valve Fundamentals. Valves are common components in nuclear plant piping systems, and many of them are custom built to respond to specific design and accident scenarios. The example considers a targeted market segment consisting of automatically actuated gate valves such as those shown in Fig. 2. Gate valves are used to isolate flow, and they can accomplish this better than most other valve types because (1) they are reliable due to their simple design, (2) they require less actuator force while closing

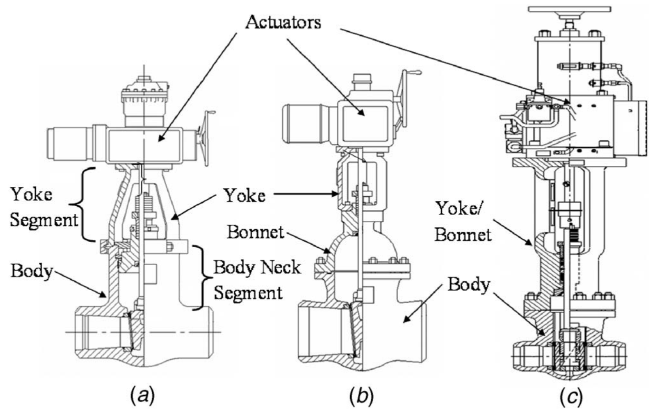


Fig. 2 Typical gate valves (courtesy of flowserve Corporation): (*a*) Size 6, Class 900 flex wedge, (*b*) Size 8, Class 150 flex wedge, and (*c*) Size 4, Class 150 double disk

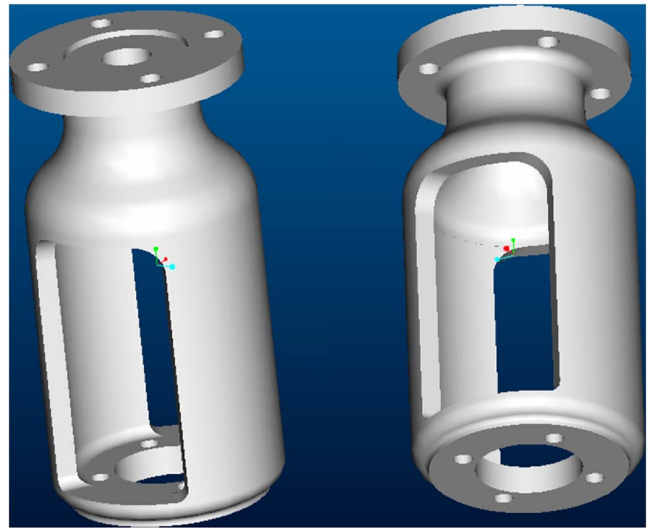


Fig. 4 Solid model views of a typical yoke

against flow, (3) they introduce minimal flow resistance while open, and (4) differential pressure across the gate aids in sealing off flow.

As an example, Fig. 3 shows a flex wedge gate in the closed position. Flow isolation and sealing are achieved through bearing contact between the gate and the seat ring that is welded into the valve body. The actuator provides thrust to the gate and must provide enough force to overcome frictional and flow induced drag and to wedge the gate into the seat. Once closed, differential pressure can develop, which forces the gate against the seat on the downstream side of the valve. Then, both differential pressure and wedging forces are available to affect a seal between the gate and seat. Often, differential pressure alone provides adequate seat bearing stress to seal. Due to design symmetry, a flex wedge gate valve is bidirectional in that it can isolate flow moving in either direction.

The actuator can be manually, electrically, pneumatically, or hydraulically energized, and its function on a gate valve is to open and close the valve by raising and lowering the stem. The yoke is one of the valve components that often requires modification to respond to specific customer requirements such as loading associated with an anticipated seismic event, sensor or control installation, and mounting and support for the specific actuator size and type. As shown in Fig. 4, the yoke consists of top and bottom mounting flanges joined by two legs. This example has a transition neck between the legs and the actuator mounting flange.

In our previous work [34], we developed a product platform portfolio for the yokes that mount the automatic actuators on the nuclear grade gate valves using the product platform concept exploration method (PPCEM) [35]. The step-by-step application of the PPCEM is demonstrated through the creation of product platforms consisting of modular, scalable valve yoke cross sections for use on the gate valves. First, the market segmentation grid is constructed based on past sales data that sets a target range of valve designs for the yoke redesign. A set of design parameters is established that reasonably meet the past requirements of the targeted segment to benchmark against the redesigned yoke cross-section platforms. Parameters include actuator weight and center of gravity, yoke construction material, yoke height, actuation load, seismic load, allowed natural frequency of vibration, allowable stress criteria, and operating temperature. In addition, a yoke casting pattern modular architecture has been proposed, as shown in Fig. 5.

In subsequent work [36], we introduced a prototypical web-based specification system for the yokes. The platform leveraging strategy was refined, more platforms were created, and a body component was included to demonstrate how an entire valve external structure can be addressed.

Figure 6 shows the resulting leveraging strategy, which em-

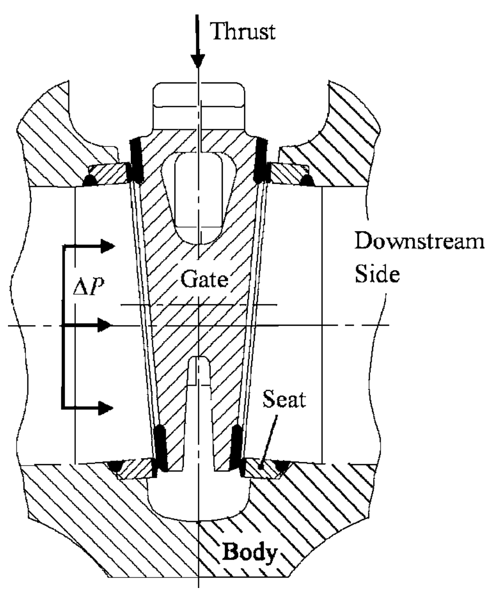


Fig. 3 Flex wedge gate valve sealing

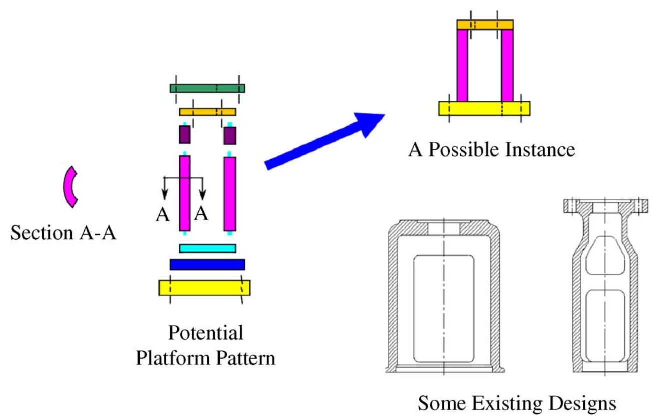


Fig. 5 Modular platform pattern concept for yoke cross sections [34]

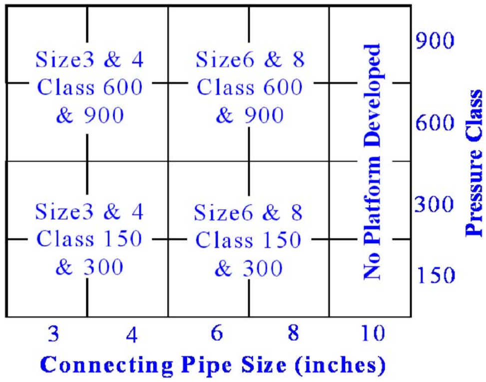


Fig. 6 Previous market segmentation grid leveraging [34]

Table 1 Sample of market segment grid artifact ordinals

		Size					
Type	Class	3	4	6	8	10	12
DD	150	1	2	3	4	5	6
	300	7	8	9	10	11	12
	600	13	14	15	16	17	18
	900	19	20	21	22	23	24
	1500	25	26	27	28	29	30
FW	150	31	32	33	34	35	36
	300	37	38	39	40	41	42
	600	43	44	45	46	47	48
	900	49	50	51	52	53	54
	1500	55	56	57	58	59	60

ploys horizontal leveraging on two characteristics (size and class). In addition, horizontal leveraging is employed across two types of gate valves: the flex wedge and the double disk. Note that this leveraging strategy was chosen simply by judgment, without consideration of optimal cost effectiveness.

This previous work forms the basis for the market segmentation grid and the platform design strategy employed in the example in Sec. 3.2. Here, we add one size (Size 12) and one class (Class 1500) to the grid, and apply the proposed method in an attempt to achieve optimal leveraging. In addition, the underlying platform optimization problem is refined such that platform solutions are obtained that better satisfy design criteria optimally. It is important to realize that (1) a bottom-up platform approach [34] is employed in that an existing valve line is redesigned to improve commonality, and (2) the subject valve line involves low volume demand yet can involve very diverse design requirements.

Then, the artifact addressed in this example is the valve, and the common, but uniquely designed, component is the yoke leg for each valve. The existing artifacts contain yoke legs with variously shaped cross sections, and it is desired to design the legs based on the common shape shown in Fig. 7. The figure shows the design variables, a, b, and c, that affect critical performance, including valve fundamental natural frequency of vibration and yoke leg stress. The valve market segmentation grid is defined by type, consisting of double disk (DD) gates and flex wedge (FW) gates, pressure class, which relates to the allowable working pressure of the process fluid, and size, which relates to the connecting piping nominal size.

A realistic goal is to design the top and bottom interface flanges for attachment of a component platform yoke to multiple artifacts in addition to the design of yoke leg cross sections. This could be accomplished by incorporating a flange design cost model into the trade-off metric resulting in a balance between the cost of interfacing a component platform with multiple artifacts and the savings from component reuse. However, to demonstrate clearly the four-step method without unnecessary complexity, we only consider commonality from yoke leg cross-section reuse. As noted earlier when describing Step 3, we assume that maximizing commonality minimizes costs, and therefore, the example portfolio optimization problem of Eq. (2) simplifies to minimizing N subject to the bounds on X^p. Despite the simplification, the resulting solution gives interesting insight regarding maximum achievable commonality, and is a valuable and necessary first step toward a real-world solution.

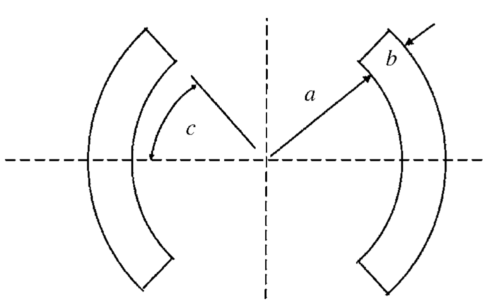


Fig. 7 Generalized yoke leg cross section defining design variables *a*, *b*, and *c*

3.2 Example Product Platform Portfolio Optimization. The example uses Microsoft Excel™ to perform calculations and store platform design parameters and a Microsoft Access™ database to track optimization statistics. A single workbook contains a spreadsheet of parameters for each member of the market segment grid, and other workbooks contain macros used to access performance and to conduct the optimizations. Table 1 is a pivot table from the database that lists grid-defining parameters and shows the ordinals that define the valve artifact members. As can be seen, the grid consists of a total of 60 different valves.

Step 1: Optimal yoke leg cross sections. Each artifact's yoke leg cross-section size (defined by the design variables a, b, and c) was optimized using Excel's solver add-in. We desire the resulting cross-section dimensions to be of a specified precision, i.e., lengths in 8th in. increments and angles in 5 deg increments, and the integer programming capabilities of the Excel solver add-in make this possible. The optimization problem for a single artifact i $(i=1,\ldots,60)$ is stated as follows:

$$\text{Minimize } F_i = \{A_i(X_i) + (f_{1,i}(X_i) + f_{2,i}(X_i)) - (\sigma_{1,i}(X_i) + \sigma_{2,i}(X_i))\} \tag{3}$$

$$X_i = (a_i, b_i, c_i)$$

Subject to

$$P_{1,i} = (1 - f_{1,i}(X_i)/f_{\min}) \le 0$$

$$P_{2,i} = (1 - f_{2,i}(X_i)/f_{\min}) \le 0$$

$$P_{3,i} = (\sigma_{1,i}(X_i)/S_{\max} - 1) \le 0$$

$$P_{4,i} = (\sigma_{2,i}(X_i)/S_{\max} - 1) \le 0$$

$$B_{1,i} = a_{\min,i} - a_i \le 0$$

$$B_{2,i} = a_i - a_{\max,i} \le 0$$

$$B_{3,i} = a_i + b_i - r_{\max,i} \le 0$$

$$X_i \ge 0$$

$$i = 1, \ldots, 60$$

The analyses relating each of the design variables (a_i, b_i, c_i) to the cross-sectional area $A_i(X_i)$, natural frequencies $f_{1,i}(X_i)$ and

Table 2 Five examples of Step 2 testing for Artifact 15. For artifact 15, $a_{min}=3.442$, a_{max} is not applicable, and $r_{max}=7.308$ as noted in Appendix A.

Component ordinal, i			1	2	3	4	5
Component design variables, X_i		a	5.5	2.625	3.5	4.75	3.125
		b	0.625	0.75	1.125	0.625	1
		c	0.5236	0.5236	0.5236	0.5236	0.6109
Artifact constraints, B_{1-3}, P_{1-4}	Step 2a	B_1	−2.058	0.817	−0.058	−1.308	0.317
		B_2			Not applicable		
		B_3	−1.183	−3.933	−2.683	−1.933	−3.183
	Step 2b	P_1	−0.371	Not required	−0.2697	−0.1798	Not required
		P_2	0.0507		−0.2354	0.3177	
		P_3	−0.7236		−0.5194	−0.5452	
		P_4	−0.8667		−1.012	−0.7415	
Test result			1	0	2	1	0

$f_{2,i}(X_i)$, and yoke leg stresses $\sigma_{1,i}(X_i)$ and $\sigma_{2,i}(X_i)$ have been published elsewhere [34] and are therefore not repeated here. For each artifact, the natural frequency constraint, f_{min}, is 33 Hz, and the maximum allowable stress limit S_{max} is 26.25 ksi. Meanwhile, the bound constraints on the design variables B_i ensure proper fit of the yoke on the corresponding artifact and vary as summarized in the table in Appendix A. Fit is governed by the size and type of connection such as a flanged or clamped connection, and the optimized yoke must conform to the artifact's existing connection design.

By solving the optimization problem for each artifact, a set of 60 candidate component platforms results, one platform for each artifact, and any resulting product platform portfolio is a subset of these. The goal of each optimization is to minimize yoke leg cross-sectional area, which effectively minimizes raw material cost. The presence of the performance parameters in the objective function forces the solution to satisfy the performance constraints as close as possible to their limits by effectively adjusting the cross-section's two-dimensional radius of gyration for required strength with minimum area, which matches the intent of an expert valve designer. In addition, yoke design consistency is introduced among the 60 artifacts in that all optimized yoke cross sections have a common shape, and this increases the leveraging potential of instantiating a single candidate platform yoke for several artifacts.

Step 2: Feasibility testing. For any given artifact, only the candidate yoke component platforms that satisfy performance and bound constraints for that artifact are feasible platforms for instantiation on that artifact. In Step 2, each of the 60 candidate yoke component platforms is tested for potential (i.e., feasible) instantiation on each of the 60 artifacts. As proposed in the method, the testing is conducted in two steps.

Step 2, Phase 1: Bound feasibility test. During the optimization process in Step 1, optimal yoke leg parameters (a_i, b_i, and c_i) and bound constraints ($B_{1,i}$, $B_{2,i}$, and $B_{3,i}$) for the 60 valves are stored in a database table for use in this step. For the first test phase, each member is tested in turn for bound constraint feasibility with cross-section parameter input equal to the optimal parameters from all other members. Each member requires 59 tests (the member's own optimal parameters are feasible by definition), and so a total of 3540 tests are required ($=60^2-60$). If the bound constraints are satisfied, then that cross section represents a candidate platform, and a "1" is assigned to a test result variable; otherwise, "0" is assigned. A star "*" is assigned to the test variable for a member's self-parameter test. For each member, test results are stored in a character string of length 60, and the sequential row upon row combined results for the 60 valve members yields a square matrix of dimension 60, with 1's or zeros off diagonal, and stars along the diagonal. Although many tests are required (3540), this substep takes little time because reformulation of the optimization problem is not required.

Step 2, Phase 2: Performance feasibility test. In Phase 2, each candidate (i.e., those corresponding to a 1 from the matrix) is tested to determine if the performance constraints ($P_{1,i}$–$P_{4,j}$ in Eq. (3)) are satisfied. Because performance constraints are involved, reformulation of the optimization problem is required, but only once for each artifact. Although, the reformulation can be time consuming, the test needs to be conducted over only these candidates—not every entry in the matrix. If the candidate meets the constraints, then the test matrix entry is marked with "2." Once complete, the final candidates are those marked with 2.

For this example, the resulting matrix is shown in Appendix B. Note that of the 3540 required Phase 1 feasibility tests, only 1250 tests are required during Phase 2; thus, only 35% of the candidates need to be evaluated with the more expensive analyses in Phase 2. Phase 2 testing yields 734 feasible candidates (those marked with either 2 or $*$).

Table 2 shows five examples of Step 2 feasibility tests for Artifact 15, which corresponds to the Size 6 Class 600 DD gate valve according to the numbering scheme from Table 1. These sample tests correspond to the boxed and shaded region shown in Appendix B. The table includes data to verify the given bound constraints, but notice that constraint B_2 is not applicable for this artifact (i.e., design variable a is unconstrained in the upper limit). As can be seen, candidate component Platforms 2 and 5 fail to meet the bound constraints, and thus these two candidates are no longer considered. Evaluation of the performance constraints for the remaining Candidates 1, 3, and 4 involves determining valve extended structure natural frequencies and yoke leg stress levels (see Ref. [34] for more details on these analyses). Although the stress and natural frequency calculation details for the remaining three candidates are not given, the table shows resulting performance constraint values, which adequately demonstrates the methodology. The table shows that only component Platform 3 meets all four performance requirements and earns a 2 in the matrix as the final test result. As can be seen, the testing process is straightforward, and further details are not really needed to illustrate the novel aspects of the proposed four-step method.

Step 3. Optimization problem formulation. The most innovative part of the proposed four-step method involves optimization problem formulation, and adequate details are provided so that interested readers can replicate the results. The optimization requires building candidate platform arrays from the matrix in Appendix B. Using the first row as an example, 14 cross sections satisfy all valve Ordinal 1 constraints; therefore, 14 is the size of the C_1 array as well as the upper bound on $X^P{}_1$. As another example, the last matrix row 60 has four feasible cross sections such that the upper bound for $X^P{}_{60}$ is 4, and the candidate array (C_{60}) is {29, 30,

Table 3 Component platform portfolio Solution 1

Type	Class	Size 3	4	6	8	10	12	Ordinal	Cross Section	Qty
DD	150	23	37	48	29	48	30	23	10-900-DD	7
	300	29	29	29	23	47	23	29	10-1500-DD	11
	600	37	48	29	47	30	30	30	12-1500-DD	10
	900	23	37	29	23	23	30	37	3-300-FW	9
	1500	37	23	30	30	29	30	44	4-600-FW	3
FW	150	37	37	48	48	47	29	47	10-600-FW	8
	300	37	37	29	48	48	47	48	12-600-FW	12
	600	37	44	48	47	47	48			
	900	44	44	48	47	30	30			
	1500	48	47	29	48	29	30			

Table 4 Component platform portfolio Solution 2

Type	Class	Size 3	4	6	8	10	12	Ordinal	Cross Section	Qty
DD	150	47	20	51	26	51	47	20	4-900-DD	5
	300	26	26	26	26	47	47	26	4-1500-DD	12
	600	37	26	26	30	30	30	30	12-1500-DD	13
	900	47	20	26	47	30	30	37	3-300-FW	5
	1500	37	26	30	30	30	30	45	6-600-FW	4
FW	150	20	20	45	45	26	47	47	10-600-FW	13
	300	37	20	45	26	26	47	51	6-900-FW	8
	600	37	51	45	47	47	51			
	900	37	51	51	47	30	30			
	1500	51	47	47	51	30	30			

59, 60}; notice that the valve's own cross section is included (60 in this case), which should always be a condition.

The first three columns of the table given in Appendix C give candidate arrays and the maximum allowed X^P values for each artifact, which is the complete input required to set up the platform portfolio optimization problem. As discussed previously, minimizing N is the objective (to maximize commonality), where N is computed as the number of unique members of $C(X^P)$, where $C(X^P)$ defines the component platform ordinals used to span the market segments. The next step is to solve the optimization problem.

Step 4. Solving the optimization problem. The optimization is solved using a SA algorithm that is presented in Ref. [37]. The starting temperature was 10,000, the temperature reduction fraction was 0.7, the number of temperature reductions was 5, the number of search cycles per temperature reduction was 120, and the convergence criterion was set to 0.0001. The SA algorithm consistently converges to a feasible solution, but the resulting optimal number of component platforms (N^*) varies from 7 to 12. More than 50 different runs were performed before declaring $N^*=7$ the maximum achievable commonality. The nature of this inconsistency requires additional investigation; however, it does not diminish the utility of the proposed four-step method since all of the solutions offer improved commonality over the existing product line (i.e., from 60 different yoke legs for each artifact to 7 component platforms).

In addition to the inconsistency, use of the SA algorithm is computationally expensive, but the expense is offset by the simplicity of the example portfolio objective function. One run for this example takes about 10 min on a 1.8 GHz PC running Windows XP Professional™ and requires about 1,500,000 function calls, which could be reduced by using less conservative parameter settings in the SA. For comparison, the number of function calls required by an exhaustive search equals the product of all the upper bounds of the X^P array, which, for this example, is greater than $2.7E+58$ evaluations based on the maximum X^P column in Appendix C. Additional work is needed to reduce the number of function calls by the SA algorithm, in particular, and for solving this optimization problem, in general. Parallel computing techniques could also be employed as the problem is easily parallelizable for different market segments.

The last two columns from the table in Appendix C give two $N^*=7$ optimal solutions, where the optimal X^P array and the corresponding used platforms $C(X^P)$ are given. In addition, Tables 3 and 4 contain pivot tables on the left that graphically define the solutions, and the numbers in the tables correspond to the unique source component platform ordinals and descriptions listed on the right. The tables help reveal patterns: For instance, both solutions employ Platforms 30, 37, and 47, which are shown shaded, and Platform 30 can be used in at least ten different artifacts, which represents a significant gain in commonality.

In addition to the multiple solutions noted earlier, the corresponding platform ordinals can vary widely between solutions; however, all of the results are feasible. A good reason for the varied results is this: Given that a certain platform is suitable for a given valve, that valve's cross section may be suitable for the platform's valve and also for many of the valves that use the platform. In other words, the platforms can have a "reflexive" property in that it may be possible to switch a used platform for the platform of one of its users. At first glance, some of the leveraging apparent in the solutions does not seem viable. For instance, use of Platform 48 (from the Size 12, Class 600 FW gate valve) on Valve 55 (the Size 3, Class 1500, FW gate valve) seems discrepant; however, this is a verified possibility. A reason for the discrepancy is that the underlying valve artifacts are currently lacking in commonality, as no smooth transition in yoke mounting parameters exists among valve sizes, pressure classes, or types.

4 Closing Remarks

The proposed four-step product platform portfolio optimization method shows promise for determining a component platform mix that maximizes commonality within an existing product line. The given example involving redesign of a single component that is used by all members of an existing product line is the first attempt at employing the proposed method. The example involves an existing market segmentation grid that is fully populated, and the entire grid is addressed; however, the method can be applied to a partially full grid as well. For instance, the procedure could be used to redesign a component that is common to only a portion of the artifacts in a market segmentation grid, and these artifacts could have random placement within the grid. Although the present method may be limited to problems similar to the example, variants may emerge for the general design or redesign of multiple components in an emerging or existing product platform portfolio.

Trade-offs between cost and commonality were not explicitly considered in this example, and their consideration could improve the overall cost effectiveness of the resulting component portfolio, as the minimum number of platforms that satisfy the constraints may not be the most economical to implement. To be effective, the trade-off metric should capture the cost of using the component platform portfolio relative to the cost of using the existing unique components. The metric should consider costs due to such things as new tooling, raw material, machining, and setup, or the benefit due to such things as reduced overhead, and savings in reduced time to market. In addition, the metric should include sales volume in order to assess payback time, as in general, a component platform portfolio will not be profitable until a sufficient sales volume is realized.

Based on this, future work suggests implementing a general complementary method for constructing a trade-off metric that considers cost. Proper modeling of costs may help improve consistency of results, or further research efforts may lead to other techniques for improving consistency. In addition to the solution inconsistency challenge, other potential work is to develop an algorithm that explores changing chosen platform parameters such that fewer platforms are required yet portfolio performance is not compromised.

Acknowledgment

This work was supported in part by the National Science Foundation under CAREER Award No. DMI-0133923.

Nomenclature

a = yoke leg cross-section inside radius (design variable)
A = yoke leg cross-section area
b = yoke leg cross-section thickness (design variable)
B_i = boundary constraints for grid member i
c = yoke leg cross-section angle (design variable)
C_j = candidate platforms for grid member i
f_1, f_2 = valve primary mode natural frequencies
$f_{\min}$ = minimum allowed natural frequency
F_i^* = optimal objective function for grid member i
i = market segment grid member ordinal
j = optimal component ordinal for grid member i
n = number of market segment grid members
N = number of used platforms
N^* = minimum required number of used platforms
P_i = performance constraints for grid member i
S = market segment grid ordinal array
S_A = allowable stress
T = trade-off metric for platform portfolio optimization
X_i = candidate component platform design variables for grid member i
X_i^* = optimal component platform solution for grid member i
X^P = platform portfolio design variables
X^{P*} = optimal platform portfolio design variables
σ_1, σ_2 = yoke leg critical stresses

Appendix A: Step 1, Artifact-Specific Constraints

Candidate artifact #	$a_{\min}$	$a_{\max}$	$r_{\max}$
1	5.4345	7.818	n/a
2	1.5	3	n/a
3	3.0875	4.85	n/a
4	4.7125	6.475	n/a
5	3.0665	5.141	n/a
6	5.8595	8.243	n/a
7	3.547	5.0005	n/a
8	3.672	5.1255	n/a
9	3.922	5.3755	n/a
10	4.7125	6.475	n/a
11	5.504	7.5785	n/a
12	5.5875	7.35	n/a
13	1.8125	n/a	4.3125
14	2.665	n/a	6.775
15	3.442	n/a	7.308
16	5.14	n/a	9.86
17	5.69	n/a	11.43
18	6.5635	9.5665	n/a
19	2.005	n/a	7.245
20	1.625	3.125	n/a
21	3.573	n/a	6.927
22	5.5875	7.35	n/a
23	5.82	n/a	12.56
24	7.773	10.467	n/a
25	2	3.5	n/a
26	4.6915	6.766	n/a
27	5.701	8.0845	n/a
28	6.5665	8.641	n/a
29	4.81	n/a	11.07
30	4.125	n/a	16.125
31	1.25	2.75	n/a
32	1.3125	2.8125	n/a
33	2.4625	4.225	n/a
34	3.107	5.4905	n/a
35	3.4795	6.793	n/a
36	3.9415	6.016	n/a
37	1.375	2.875	n/a
38	1.5	3	n/a
39	3.3375	5.1	n/a
40	3.3375	5.1	n/a
41	3.0665	5.141	n/a
42	5.5875	7.35	n/a
43	2.16	n/a	4.02
44	2.4065	n/a	5.0305
45	3.8435	n/a	6.0935
46	3.7655	n/a	7.7035
47	5.82	n/a	9.06
48	2.625	4.125	n/a
49	2.062	n/a	4.188
50	2.595	n/a	5.215
51	3.5	n/a	7
52	4.31	n/a	8.57
53	5.375	n/a	10.375
54	5.815	n/a	11.315
55	2.6875	n/a	6.4375
56	2.82	6.016	7.28
57	2.594	n/a	8.718
58	3.1875	4.6875	n/a
59	3.56	n/a	12.82
60	4.115	n/a	14.155

Appendix B: Step 2, Example, Feasibility Test Matrix for Yoke Leg Example

```
*00002000022000220000220002200000000000000200002000000202000
0*0000000000220000220000200000112000210000120000020000200000
00*11011110000100000200001000000000110011100012021022000022220
200*0200022200000200000200200200002000000000000200000220200220
0021*011110000020000002000020020000001200112000220210220000222
11212*11111211122211222212222211121211111211222211222221222
000200*2220000020000002000020020000022002220002202000200000220
0002000*22000020000000000020020000002200222000220200020000022
00020000*200002000000000000200200000222002220000002000220000022
```

200202000*22000020000020020020000200000000000020000022020022
0000020000*2000220000220002200000000000002000020000002020000
00000100001*0002200002200012000000000000002000020000002010000
010020100000*10000110000100000111000210000110000110000200000
2022221112222*2000002000120020001212001220002202222200222200
10210211111200*020002020020020000212001120002222002220010220
100001000011000*20000110001102000100000001000020000022010002
0000010000010001*200011200110200000000000010000200000002000000
00000000000000000*000100001102000000000000000000000000000000
212122111122212020*12020120020111212211120112222122220222220
0200200000002200002*0000200000112000210000120000020000202000
1001011111200100000*000020020000111001110002202000220010220
0000010000120002200000*2000120000000000000020000200000002010000
0000010000000002020002*20012020000000000002000020000002000000
000000000000000000100000*000002000000000000000000000000000000
0120200000002200002100000*00000111000210000120000222000202000
100101000112000220000220000*00200001000000020000200000022010022
00000100000200022000022000*20200000000000020000200000002000000
0000000000000000000000100001*020000000000000000000000000000000
10000100001100011100011100011*20001000000010000100000011010001
100101000111000111000111011111*000111001001000011000111010011
02000000000002000022000000000*200002200002200000000000200000
020000000000020000220000000002*00002200002200000000000200000
0120202220002220002020002000000*000010220122202222000202200
00211011110000100000200002002000*12001110002202102220002222
20220211122200222000222002002000002*20012220002222002222020222
100101001112001020000000002002000021*001110000022000222010022
01000000000021000011000000000001100000*10000110000000000200000
02000000000022000022000020000011200002*00002200000020000200000
00220011120000200000200002002000002200*2200022020022000000220
2001020011220020200000000020020000212001*20000022000222020022
00212011110000200000200002002000001200011*0002202102200002220
000001000011000220000110001200000000000000*000020000002010000
0200000000002200002200002000001120002200000*200002200000000000
0120201110002220002020001000000010000112201*2000222000202000
00010000110000200000000002000000001200112000*202000200000000
100101001111001220000110011020000111001111001*21000220010022
0000010000000001000001100011000000000000000010000*0000002000000
01101011100011100000100010000000100000011001120*112000102100
010020000000220000210000100000111000210000120000*20000200000
01202011100022200000200020000000102000222002200002*2000202200
10110111111100102000100001002000011100111000120200*220010220
100101000111000220000120011220000111001002000020000*22010022
10000100001100011200011000110200010000000010000100000*2010000
000001000000000101000112001102000000000000100001000000*000000
102110111110111000002000110000001111001110001202112200*12200
1021221111222020200020001200200012120011200022221122200*2220
11111111111111122000111011122000111100111201122111222211*222
00100011100000100000100000000000011001110001202102200000*00
1001011111110011110011110111220001110011110011110001110101*1
10010100011100011100011101112200011100100100001100011101002*

Appendix C: Steps 3 and 4 Example, Optimization Setup and Solution

Artifact ordinal	Candidate platforms (C)	Max X^P index	Solution 1		Solution 2	
			X^{P*}	$C(X^{P*})$	X^{P*}	$C(X^{P*})$
1	1,6,11,12,16,17,22,23,27,28,42,47,54,56	14	8	23	12	47
2	2,13,14,19,20,25,33,37,44,50,55	11	8	37	5	20
3	3,21,46,48,51,52,57,58,59	9	4	48	5	51
4	1,4,6,10,11,12,17,23,26,29,34,47,53,54,56,59,60	17	10	29	9	26
5	3,5,15,21,26,29,36,41,45,46,48,51,52,57,58,59	16	11	48	12	51
6	3,5,6,12,16,17,18,21,22,23,24,26, 27,28,29,30,34,36,42,45,46,47, 48,51,52,53,54,55,57,58,59,60	32	16	30	22	47
7	4,7,8,9,10,15,21,26,29,35,36,39,40,41,45,46,48,52,58,59	20	9	29	8	26
8	4,8,9,10,15,26,29,35,36,39,40,41,45,46,48,52,58,59	18	7	29	6	26
9	4,9,10,15,26,29,34,35,36,39,40,41,48,52,53,59,60	17	6	29	5	26

Artifact ordinal	Candidate platforms (C)	Max X^P index	Solution 1		Solution 2	
			X^{P*}	$C(X^{P*})$	X^{P*}	$C(X^{P*})$
10	1,4,6,10,11,12,17,23,26,29,34,47,53,54,56,59,60	17	8	23	9	26
11	6,11,12,16,17,22,23,27,28,42,47,54,56	13	11	47	11	47
12	12,16,17,22,23,28,42,47,54	9	5	23	8	47
13	5,13,37,55	4	3	37	3	37
14	1,3,4,5,6,10,11,12,13,14,15,21, 26,29,34,36,40,41,45,46,48, 49,50,51,52,55,56,57,58	29	21	48	13	26
15	3,6,12,15,17,21,23,26,29,34,36,41,45,46,47,48,51,52,53,58,59	21	9	29	8	26
16	16,17,30,47,53,54,60	7	4	47	3	30
17	17,18,24,30,47,54	6	4	30	4	30
18	18,30	2	2	30	2	30
19	1,3,5,6,11,12,13,15,17,19,21,23, 26,29,34,36,37,41,45,46,47,48, 50,51,52,53,55,56,57,58,59	31	12	23	21	47
20	2,5,13,14,19,20,25,33,37,44,50,55,57	13	9	37	6	20
21	12,21,26,29,45,46,48,52,53,58,59	11	4	29	3	26
22	12,16,17,22,23,28,42,47,54	9	5	23	8	47
23	16,18,22,23,24,28,30,42,47,54	10	4	23	7	30
24	24,30	2	2	30	2	30
25	3,5,13,14,19,25,37,44,49,50,51,55,57	13	7	37	7	37
26	12,16,17,22,23,26,29,42,47,53,54,59,60	13	5	23	6	26
27	12,16,17,22,23,27,28,30,42,47,54	11	8	30	8	30
28	28,30	2	2	30	2	30
29	29,30	2	1	29	2	30
30	30	1	1	30	1	30
31	2,14,19,20,31,32,37,38,43,44,55	11	7	37	4	20
32	2,14,19,20,31,32,37,38,43,44,55	11	7	37	4	20
33	3,5,7,8,9,13,14,15,19,21,25, 33,40,41,44,45,46,48,49,50, 51,55,57,58	24	18	48	16	45
34	3,21,26,29,34,36,45,46,48,51,52,53,57,58,59,60	16	9	48	7	45
35	1,3,4,6,10,11,12,15,16,17,21, 22,23,26,29,34,35,36,40,41, 42,45,46,47,48,51,52,53,54, 56,58,59,60	33	24	47	14	26
36	12,17,26,29,34,36,47,48,52,53,54,59,60	13	4	29	7	47
37	13,37,55	3	2	37	2	37
38	2,13,14,19,20,25,33,37,38,43,44,50,55	13	8	37	5	20
39	3,4,10,15,21,26,29,35,36,39,40,41,45,46,48,51,52,58,59	19	7	29	13	45
40	1,6,11,12,15,17,26,29,34,36,40,41,47,48,52,53,54,56,59,60	20	14	48	7	26
41	3,5,15,21,26,29,36,41,45,46,48,51,52,57,58,59	16	11	48	5	26
42	16,17,28,42,47,54	6	5	47	5	47
43	2,13,14,19,20,25,33,37,38,43,44,49,50	13	8	37	8	37
44	3,5,13,14,15,19,21,40,41,44,45,49,50,51,55,57	16	10	44	14	51
45	15,26,36,41,45,46,48,52	8	7	48	5	45
46	16,17,29,46,47,52,53,59,60	9	5	47	5	47
47	47,54	2	1	47	1	47
48	46,48,51,57	4	2	48	3	51
49	5,13,14,19,37,44,49,50,55	9	6	44	5	37
50	3,5,13,14,15,21,25,35,39,40,41,44,45,49,50,51,55,57,58	19	12	44	16	51
51	17,29,46,48,51,52,53,58,59	9	4	48	5	51
52	16,17,23,28,29,42,47,52,53,54,59,60	12	7	47	7	47
53	18,30,53,54	4	2	30	2	30
54	24,30,54	3	2	30	2	30
55	3,21,46,48,51,52,55,57,58	9	4	48	5	51
56	3,5,6,11,12,13,15,17,21,26,29, 34,36,41,45,46,47,48,51,52, 53,56,57,58,59	25	17	47	17	47
57	16,17,28,29,42,46,47,51,52,53,54,57,58,59,60	15	4	29	7	47
58	46,48,51,52,58	5	2	48	3	51
59	29,30,59	3	1	29	2	30
60	29,30,59,60	4	2	30	2	30

References

[1] Sanderson, S. W., and Uzumeri, M., 1997, *Managing Product Families*, Irwin, Chicago.

[2] Meyer, M. H., 1997, "Revitalize Your Product Lines Through Continuous Platform Renewal," Res. Technol. Manag., **40**(2), pp. 17–28.

[3] Pessina, M. W., and Renner, J. R., 1998, "Mass Customization at Lutron Electronics-A Total Company Process," Agility & Global Competition, **2**(2), pp. 50–57.

[4] Aboulafia, R., 2000, "Airbus Pulls Closer to Boeing," Aerosp. Am., **38**(4), pp. 16–18.

[5] Dahmus, J. B., Gonzalez-Zugasti, J. P., and Otto, K. N., 2000, "Modular Product Architecture," *ASME Design Engineering Technical Conferences-Design Theory and Methodology Conference*, Baltimore, MD, ASME Paper No. DETC2000/DTM-14565.

[6] 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 Paper No. DETC-2000/DAC-14238.

[7] 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.

[8] Ulrich, K., 1995, "The Role of Product Architecture in the Manufacturing Firm," Res. Policy, **24**(3), pp. 419–440.

[9] Zamirowksi, E. J., and Otto, K. N., 1999, "Identifying Product Portfolio Architecture Modularity Using Function and Variety Heuristics," *ASME Design Engineering Technical Conferences-Design Theory and Methodology Conference*, Las Vegas, NV, ASME Paper No. DETC99/DTM-8760.

[10] Messac, A., Martinez, M. P., and Simpson, T. W., 2002, "Effective Product Family Design Using Physical Programming," Eng. Optimiz., **34**(3), pp. 245–261.

[11] 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.

[12] Simpson, T. W., Seepersad, C. C., and Mistree, F., 2001, "Balancing Commonality and Performance Within the Concurrent Design of Multiple Products in a Product Family," Concurr. Eng. Res. Appl., **9**(3), pp. 177–190.

[13] Dai, Z., and Scott, M. J., 2006, "Effective Product Family Design Using Preference Aggregation," ASME J. Mech. Des., **128**(4), pp. 659–667.

[14] 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.

[15] Seepersad, C. C., Mistree, F., and Allen, J. K., 2002, "A Quantitative Approach for Designing Multiple Product Platforms for an Evolving Portfolio of Products," *ASME Design Engineering Technical Conferences-Design Automation Conference*, Montreal, Canada, ASME Paper No. DETC02/DAC-34096.

[16] D'Souza, B. S., and Simpson, T. W., 2002, "A Genetic Algorithm Based Method for Product Family Design Optimization," *ASME 2002 Design Engineering Technical Conferences-Design Automation Conference*, Montreal, Canada, ASME Paper No. DETC02/DAC-34106.

[17] Simpson, T. W., Chen, W., Allen, J. K., and Mistree, F., 1997, "Designing Ranged Sets of Top-Level Design Specifications for a Family of Aircraft: An Application of Design Capability Indicies," *SAE World Aviation Congress and Exposition*, Anaheim, CA, Paper No. AIAA 97–5513.

[18] Nayak, R. U., Chen, W., and Simpson, T. W., 2000, "A Variation-Based Method for Product Family Design," *ASME 2000 Design Engineering Technical Conferences-Design Automation Conference*, Baltamore, MD, ASME Paper No. DETC2000/DAC-14264.

[19] de Weck, O. L., Suh, E. S., and Chang, D., 2003, "Product Family and Platform Portfolio Optimization," *ASME Design Engineering Technical Conferences-Design Automation Conference*, Chicago, IL, ASME Paper. No. DETC2003/DAC-48721.

[20] 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 Paper No. DETC2001/DAC-21058.

[21] Fujita, K., 2002, "Product Variety Optimization Under Modular Architecture," Comput.-Aided Des., **34**(12), pp. 953–965.

[22] Fellini, M. K., Papalambros, P. Y., and Perez-Duarte, A., 2002, "Platform Selection Under Performance Loss Constraints in Optimal Design of Product Families," *ASME 2002 Design Engineering Technical Conferences-Design Automation Conference*, Montreal, Canada, ASME Paper No. DETC2002/DAC-34099.

[23] Georgiopoulos, P., Fellini, M. K., Sasena, M., and Papalambros, P. Y., 2002, "Optimal Design Decisions in Product Portfolio Valuation," *ASME Design Engineering Technical Conferences-Design Automation Conference*, Montreal, Canada, ASME Paper No. DETC2002/DAC-34097.

[24] Hernandez, G., Allen, J. K., and Mistree, F., 2002, "Design of Hierarchic Platforms for Customizable Products," *ASME Design Engineering Technical Conferences-Design Automation Conference*, Montreal, Canada, ASME Paper No. DETC2002/DAC-34095.

[25] Mather, H., 1995, "Product Variety—Friend or Foe?," *38th American Production and Inventory Control Society International Conference and Exhibition*, Orlando, FL, APICS, pp. 378–381.

[26] Martin, M. V., and Ishii, K., 1997, "Design for Variety: Development of Complexity Indices and Design Charts," *ASME Design Engineering Technical Conferences-Design for Manufacturing Conference*, Sacramento, CA, ASME Paper No. DETC97/DFM-4359.

[27] Nelson II, S. A., Parkinson, M. B., and Papalambros, P. Y., 2001, "Multicriteria Optimization in Product Platform Design," ASME J. Mech. Des., **123**(2), pp. 199–204.

[28] Fellini, R., Kokkolaras, M., and Papalambros, P., 2005, "Commonality Decisions in Product Family Design," *Product Platform and Product Family Design: Methods and Applications*, T. W. Simpson, Z. Siddique, and J. Jiao, eds., Springer, New York, pp. 157–185.

[29] Balabanov, V., Giunta, A. A., Golovidov, O., Grossman, B., Mason, W. H., and Watson, L. T., 1999, "Reasonable Design Space Approach to Response Surface Approximation," J. Aircr., **36**(1), pp. 308–315.

[30] Knill, D. L., Giunta, A. A., Baker, C. A., Grossman, B., Mason, W. H., Haftka, R. T., and Watson, L. T., 1999, "Response Surface Models Combining Linear and Euler Aerodynamics for Supersonic Transport Design," J. Aircr., **36**(1), pp. 75–86.

[31] Simpson, T. W., 2005, "Methods for Optimizing Product Platforms and Product Families: Overview and Classification," *Product Platform and Product Family Design: Methods and Applications*, Simpson, T. W., Z. Siddique, and J. Jiao, eds., Springer, New York, pp. 133–156.

[32] 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.

[33] Khajavirad, A., and Michalek, J. J., 2007, "An Extension of the Commonality Index for Product Family Optimization," *ASME Design Engineering Technical Conferences-Design Automation Conference*, ASME, Las Vegas, NV, Paper No. DETC2006-DAC35605.

[34] Farrell, R. S., and Simpson, T. W., 2003, "Product Platform Design to Improve Commonality in Custom Products," J. Intell. Manuf., **14**(6), pp. 541–556.

[35] 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.

[36] Farrell, R. S., Simpson, T. W., Stump, G., and Park, J., 2007, "Prototype Implementation of a Virtual Product Family Through a Web-Based Custom Product Specification System," International Journal of Mass Customization, **2**(1/2), pp. 161–178.

[37] Belegundu, A. D., and Chandrupatla, T. R., 1999, *Optimization Concepts and Applications in Engineering*, Prentice-Hall, Upper Saddle River, NJ.