Research in Engineering Design (2018) 29:547–563
https://doi.org/10.1007/s00163-018-0288-5

ORIGINAL PAPER

Product family platform selection using a Pareto front of maximum commonality and strategic modularity

Kyle Baylis[1] · Guanglu Zhang[1] · Daniel A. McAdams[1]

Received: 29 May 2017 / Revised: 20 April 2018 / Accepted: 23 April 2018 / Published online: 8 May 2018

Abstract

Product family design offers a cost-effective solution for providing a variety of products to meet the needs of diverse markets. At the beginning of product family design, designers must decide what can be shared among the product variants in a family. Optimal design formulations have been developed by researchers to find one optimal component sharing solution based on commonality, cost or technical performance of a product family. However, these optimization methods may not be able to apply in consumer product design because some metrics (e.g., visual appeal and ergonomics) of a consumer product cannot be formulized. In this paper, we suggest a tradeoff between commonality and the quality of the modular architecture in product family platform selection. We introduce a method for designers to identify multiple component sharing options that lie along a Pareto front of maximum commonality and strategic modularity. The component sharing options along the Pareto front can be evaluated, compared, and further modified. We demonstrate the method using a case study of product family platform selection of high-end and low-end impact drivers and electric drills. In the case study, the quality of the modular architecture is evaluated using a design structure matrix (DSM) for each of product variants. Three architectures along the Pareto front with maximum commonality, optimal modularity, and a balanced solution of the two metrics are highlighted and further examined to validate the effectiveness of our method.

Keywords Product family design · Product platform · Modular design · Design structure matrix (DSM) · Component clustering

1 Introduction

In an effort to boost revenue, companies try to offer tailored products to meet the needs of many market segments. However, it is expensive for a company to design, manufacture, and support a large set of different products (Otto and Wood 2001). Product family design is developed to conciliate the conflict. A product family is a group of similar product variants that share one or more common components (Harlou 2006; Meyer and Lehnerd 1997). The common components form a product family platform (Erens and Verhulst 1997; Kristjansson et al. 2004). Product family design offers a cost-effective solution for providing a variety of products to meet the needs of diverse markets (Meyer and Utterback 1993; Simpson et al. 2006).

At the beginning of product family design, designers must decide what can be shared among a set of product variants aiming for different market segments. Optimal design formulations have been developed to balance commonality, cost and technical performance loss across a product family (Fellini et al. 2004; Liu et al. 2011; Nelson et al. 2000). Designers use these methods to derive one optimal component sharing solution for industrial product (e.g., automobile and airplane) development. However, these optimization methods may not be able to apply in consumer product design because a consumer product often has many other metrics for its performance, and some metrics (e.g., visual appeal and ergonomics) cannot be formulized. In this scenario, designers could exhaustively enumerate the sharing possibilities and build a chart to qualitatively evaluate these component sharing options (Otto and Wood 2001), but the

✉ Daniel A. McAdams
dmcadams@tamu.edu

Kyle Baylis
kylebaylis@tamu.edu

Guanglu Zhang
glzhang@tamu.edu

[1] Department of Mechanical Engineering, Texas A&M University, College Station, TX 77840, USA

exhaustive enumeration is only suitable for the family with a small number of sharing components.

In this paper, we introduce a method for designers to identify multiple component sharing options that lie along a Pareto front of maximum commonality and strategic modularity. The component sharing alternatives on the Pareto front can be evaluated, compared, and further modified. To our knowledge, the application of Pareto front in this manner is a new contribution to product family platform selection. Our method helps designers address the uncertainty in the design process by providing multiple component sharing options. Designers can utilize area-specific expertise in the following decision-making process. Moreover, our method is broadly applicable since the method is compatible with any commonality indices and component clustering methods. Designers also could substitute other metrics (e.g., technical performance, cost, complexity, and adaptability) for the strategic modularity used in this paper to derive one or more customized Pareto fronts.

The rest of this paper is organized as follows. We begin with a short review of background and related work of product family design in Sect. 2. The methodology to generate a Pareto front of maximum commonality and strategic modularity for platform selection is presented in Sect. 3. In Sect. 4, we demonstrate our method through a case study of an impact driver and electric drill product family platform selection in which we highlight three-component sharing options along the Pareto front of maximum commonality and strategic modularity. We conclude with the contribution of this work and future research directions.

2 Background and related work

A group of product variants that shares common components to satisfy a variety of market niches is a product family (Harlou 2006). A product platform is a group of common components that are shared by multiple products or generations of products (Kristjansson et al. 2004). Using a product platform of common components, multiple product variants can be efficiently developed (Meyer and Lehnerd 1997). Research has shown product family design to be a cost-effective option for providing variety in products, allowing designers to meet the needs of multiple market segments (Meyer and Utterback 1993; Simpson et al. 2006).

Product family design is an active research area in recent years. Researchers have developed great amount of methodologies for designers to build a product family. These methods often have a prerequisite that the platform of a product family has been selected. The specified product platform is then optimized, and product variants are designed to meet the customers' needs in different market segments. Gebhardt et al. (2014) developed a Module Interface Graph (MIG) to help designers define module interfaces and boundaries during the modularization of a product family. Lei et al. (2016) introduced additive manufacturing into product family design. By utilizing additive manufacturing, they eliminate constraints arisen in conventional product family designs from finding a compromise between commonality and technical performance. Ma and Kim (2016) developed a predictive data-driven product family design (PDPFD) method to determine optimal product family architectures with customer preference data. Wang et al. (2016) created a Stackelberg game theoretic model to optimize product family architecture. Their model considered supply chain-related issues that previously received scant research attention in product family design. Jung and Simpson (2016) employed an integrated approach that utilizes multiple product family metrics to develop an effective product platform redesign strategy. More research publications in product family design and platform-based product development before 2014 could be found in an edited book (Simpson et al. 2014) or review papers written by Jose and Tollenaere (2005), Jiao et al. (2007), Pirmoradi et al. (2014) and Simpson et al. (2006).

At the beginning of product family design, designers must make a decision on platform selection. Designers must choose what components can be shared among product variants in a product family. Each of product variants is created by an independent design team or reverse-engineered from a dominant product in the market segment. Functional model, design optimization, and exhaustive enumeration are three major approaches that help designers to select an appropriate platform in a product family.

Zamirowski and Otto united the function structures of product variants into a family function structure to represent the entire product family (Otto and Wood 2001; Zamirowski and Otto 1999). Shared components, which are called a platform module in their paper, can be defined by the shared functions of a product family. Stone et al. (2000) introduced a method to represent a functional model of a product in a quantitative manner. The quantitative functional models help designers to compute product families and customer need ratings for modules.

The functional model approaches are broadly applicable, but the approach only gives designers several platform selection heuristics rather than provide solution in physical form level. To determine the optimum physical platform in a product family, researchers consider the platform selection as a multidisciplinary optimization problem with respect to different design metrics. Gonzalez-Zugasti et al. (2000) formulated a general optimization problem for selecting product platforms and designing the product family that considers technical performance and the cost of the product family. Spacecraft design was used as a case study to illustrate their method. Fujita and Yoshida (2004) proposed a method to optimize the final profit of a product family under technical constraints. They took passenger airplane as an example to demonstrate their optimization method. Fellini et al. (2004) formulated a method to obtain optimal product family designs for maximizing commonality without exceeding user-specified bounds on technical performance. A family of automotive body side frames was used to demonstrate their optimization method. Moon et al. (2014) introduced a multi-objective particle swarm optimization approach to determine design variables for the best platform design strategy based on commonality and multiple technical performances of a product family. To demonstrate their approach, they used a family of general aviation aircraft as a case study. Eichstetter et al. (2015) used iterative Monte Carlo sampling to compute solution space of a generic optimization problem. The shared components were identified to optimize commonality in a product family, and a set of common components is computed for 13 vehicles with ten design parameters for each of the vehicles.

Of note, the product families discussed as case studies in these design optimization papers are industrial products (e.g., spacecraft, automobile, and airplane). The optimization objective is to maximize product technical performance(s) and commonality, or to minimize the cost of a product family, and the design optimization usually only provides one "optimal" component sharing solution. However, there are several other metrics designers must consider in product family design (Jiao et al. 2007), such as modularity (Ulrich 1994), complexity (Kim et al. 2016), sustainability (Kim and Moon 2017), and adaptability (Engel et al. 2017; Engel and Reich 2015). It is hard to include all these metrics as objectives in an optimization problem. In addition, some metrics (e.g., visual appeal and ergonomics) for consumer products cannot be formulized, thus the design optimization approaches stated earlier may not be able to apply in consumer product design. Otto and Wood (2001) suggested exhaustive enumeration in platform selection for consumer product family design. Designers could exhaustively enumerate the sharing possibilities and build a chart to qualitatively evaluate these component sharing options, but the exhaustive enumeration is only suitable for the family with a small number of shared components because the number of sharing possibilities becomes large quickly with more shared components in a product family.

3 Methodology

Based on the research gap discussed in Sect. 2, we introduce a method for product family platform selection with a balance of commonality and strategic modular architecture. We use a Pareto front of maximum commonality and strategic modularity to produce multiple component sharing options for a product family. These component sharing options along the Pareto front can be evaluated, compared, and further modified by designers based on their preference or other considerations (e.g., other design metrics). The proposed method can be applied not only to industrial product design but also consumer product design.

A block diagram of the flow of our methodology is shown in Fig. 1. We divide the components of a product family into groups based on the set of products between which they may be shared. Each of the groups is then clustered into modules using design structure matrix (DSM) clustering and combined to form the complete modular architectures of the product family. Further alternative modular architectures and platforming strategies are considered by decreasing the amount of sharing in the product family to move common components to groups of less commonality which contain components with which they are highly coupled. This process follows a trade-off, creating better modules at the cost of commonality in the product family.

To evaluate alternatives, we score the proposed product family architectures based on commonality and the quality of the modular architecture. Taking advantage of algorithmically based DSM clustering, a Pareto front of maximum commonality and strategic modularity can be determined. This Pareto front provides multiple component sharing options from which the final design of the product family may be chosen or further refined.

3.1 Method input

The proposed method requires two primary inputs. The first input is a list of the components that make up each of product variants in the family. The second is a DSM matrix for the product family.

The list of products and their constituent components may be encapsulated in a binary matrix, termed the product–component matrix (PCM) as shown in Eq. (1). Each row of the PCM corresponds to a product in the product family. Each

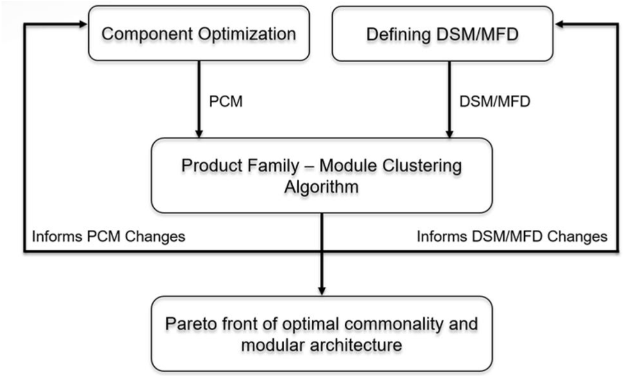

Component Optimization
Defining DSM/MFD
PCM
DSM/MFD
Product Family – Module Clustering Algorithm
Informs PCM Changes
Informs DSM/MFD Changes
Pareto front of optimal commonality and modular architecture

Fig. 1 Block diagram of the methodology flow

column corresponds to a component used in at least one of the products. A similar matrix has been introduced by Braha to partition tasks to different product development teams (Braha 2002).

$$\begin{Bmatrix} \text{product } 1 \\ \dots \\ \text{product } m \end{Bmatrix} = \begin{matrix} \text{PCM} \\ [m \times n \text{ binary matrix}] \end{matrix} \begin{Bmatrix} \text{component } 1 \\ \dots \\ \text{component } n \end{Bmatrix}. \tag{1}$$

Designers need to list the components that build the product family prior to the construction of the PCM. Determining what component each of product variants may share is an important part to set up the product–component matrix. A component may be shared if the component is functionally, morphologically, and parametrically the same in each of product variants.

3.2 Component clustering

To begin the component clustering process, we identify the platforming strategy with the highest possible commonality (given the PCM input). Components in the PCM are first organized into groups based on the set of products to which each of components is common. This grouping of components is designed around one rule: if Component A and Component B are both included in the same set of product variants, these components are placed into the same group. These organized groups are termed a platforming strategy. The platforming strategy can be visualized using a Venn diagram. Each segment of the diagram contains components that are shared by the same set of products, different than that of components in other groups. Each distinct group of components is termed as a shared group. The Venn diagram in Fig. 2 shows the shared groups of a family with three product variants.

After identifying the platforming strategy of maximum commonality, we use DSM clustering to identify the optimal modular architecture for this strategy. The design structure matrix (DSM) is a commonly used tool for defining the module boundaries of a product. DSM clustering was first introduced by Steward (1981) for designing complex systems. Since then, DSM has been adapted for grouping and organizing components into modules within a product (Eppinger and Browning 2012; Yassine and Braha 2003). Of note, there exist several other methods to cluster components into modules for a product (Holtta and Salonen 2003; Östgren 1994), such as modular function deployment (MFD) (Erixon 1998) and hybrid DSM and MFD approach (Borjesson and Hölttä-Otto 2014). Designers are free to use one of these methods to cluster components into modules.

For each of the shared groups in our platforming strategy, a new DSM is created from the DSM of the product family. The new DSM for each group contains only the components within that shared group and the interactions between the components. We then separately cluster each of the shared groups using a DSM clustering algorithm. Researchers have developed several algorithms for DSM clustering (Helmer et al. 2010; Thebeau 2001; Yu et al. 2007). Designers can choose the algorithm based on their preference. The DSM clustering modularizes each of the groups individually. Modularizing the product family in this manner produces the optimal modular architecture for a given platforming strategy.

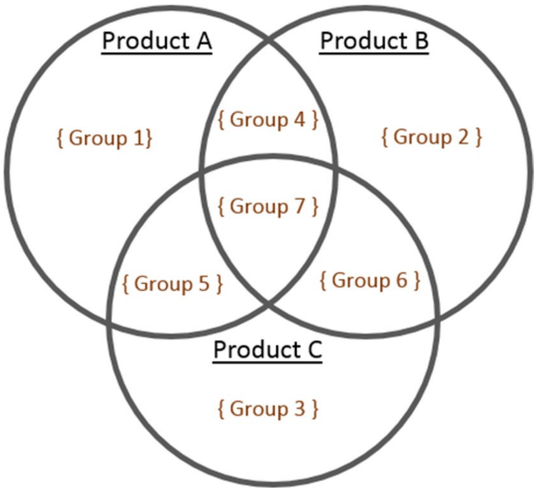


Fig. 2 Venn diagram of the shared groups of a family with three product variants

3.3 Alternative platforming strategies

Further modular architectures may be considered by decreasing commonality in the product family to identify alternative platforming strategies. A component that might be shared by a large number of product variants could be shared by a lesser number of product variants to allow the component to be integrally modularized with another group (Robertson and Ulrich 1998).

This process of decreasing commonality to attain different possibilities for modularizing the product can be represented again using a Venn diagram. The diagram in Fig. 3 demonstrates the differences in the leveraging strategies of two architectures. This process of decreasing sharing to obtain different modular architectures can be carried out for every component in a product family. Repeatedly carrying out this process of decreasing the commonality of each of the components in the product family ultimately leads to a product family with no common component. In this case, each product in the product family may be modularized independently. Thus, the optimal

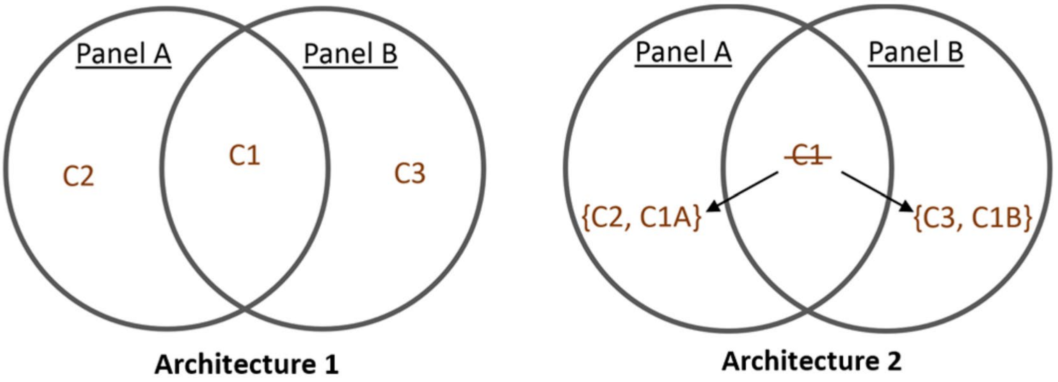


Fig. 3 Venn diagrams of the leveraging strategies of architectures 1 and 2

modular architecture for each product variant may be chosen, but the family will not receive the benefits of commonality.

3.4 Number of platforming strategies

Because of the computational nature of the described method, it is valuable to determine the number of operations required to analyze the product family. As the number of products that contain a given component increases, the number of possible platforming strategies for that component increases with its corresponding Bell number (Bell 1934), B_n. Bell numbers give the number of ways a set of elements can be partitioned into nonempty subsets, or in this case, the number of ways a component can be split into component instances shared by different sets of products. Table 1 gives the first nine Bell numbers.

Given that each of components that are common to n number of product variants may be shared in B_n number of ways, the maximum number of platforming strategies of a product family can be calculated by

$$\text{Number of platforming strategies} = \prod_{pi}^{C} B_{pi} \qquad (2)$$

where B_{pi} is the Bell number for pi, pi is the number of product variants that can share component i, and C is the number of components in the product family.

To reduce the computation time of finding optimal platforming architectures, we suggest a strategic search technique. We utilize the DSM of the product family to predetermine where shared group components should be moved to in order to improve the modular architecture of the product family. In this case, a component would only be moved into a lower level of sharing that contains component(s) with which it has DSM connections, thus eliminating a large number of platforming strategies from evaluation.

3.5 Architecture evaluation

To evaluate the large number of component sharing alternatives, we introduce the scores for commonality and modular architecture to generate a Pareto front for platforming architecture options. While the following metrics provide a good basis for assessing proposed platforming architectures, these metrics may be altered or added upon to best meet design-specific goals.

3.5.1 Commonality assessment

A number of different indices have been developed for assessing commonality between product variants within a family. The widely used commonality indices include but are not limited to degree of commonality index (DCI), total constant commonality index (TCCI), product line commonality index (PCI), percent commonality index (%C), commonality index (CI), component part commonality ($CI^{(c)}$), commonality versus diversity index (CDI), comprehensive metric for commonality (CMC), and total commonality metric (TCM). Each of the commonality indices have major focus but also have its limitations (Pirmoradi et al. 2014; Thevenot and Simpson 2006). Of note, our method is compatible with any commonality indices. Designers can choose whichever commonality index is best suited for their application to assess the commonality of the platforming architecture alternatives.

As an example, we use constant commonality index (TCCI) (Wacker and Treleven 1986) in our case study of impact

Table 1 List of Bell numbers (Bell 1934)

n	1	2	3	4	5	6	7	8	9	…
B_n	1	2	5	15	52	203	877	4140	21,147	…

drivers and electric drills family design in Sect. 4. The TCCI assesses commonality based on the number of parent products of each component in the family. The equation for calculating the TCCI is:

$$\mathrm{TCCI} = 1 - \frac{d - 1}{\sum_j^d \Phi_j - 1} \tag{3}$$

where d is the number of distinct component instances and Φ_j is the number of products to which component instance j is common. The TCCI is easy to calculate and setup (Thevenot and Simpson 2006). The TCCI provides a good initial estimate of the benefits of added commonality. However, the TCCI provides a relatively simplified view of commonality. The TCCI weights each component equally when assigning the commonality score. Therefore, it may not fully account for the complex benefits gained from sharing each of components. A more detailed representation of the benefits of commonality can be attained using an index that reflects more subtle designer preference.

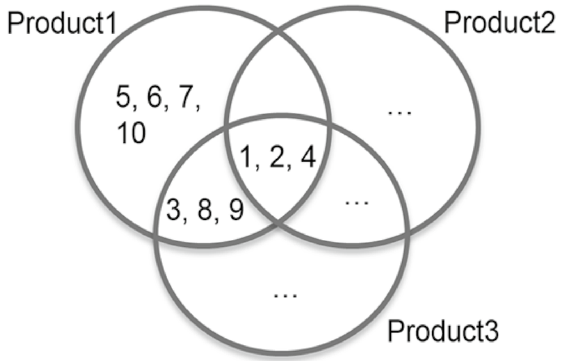
3.5.2 Modular architecture assessment

To assess the quality of a proposed modular architecture, we develop a modularity score termed as strategic modularity. The modularity score is based on the clustering cost used in DSM clustering algorithms. When using DSM clustering algorithms to find an optimal modular design, a cost is calculated for each proposed clustering of components into modules. This clustering cost is typically the sum of IntraClusterCost, or the cost of interactions occurring within a cluster/module, and ExtraClusterCost, or cost from interactions occurring between components in separate clusters/modules. DSM clustering algorithms attempt to minimize this cost function in their search for the optimal clustering, or modularization, of a product. Using the cost function of a chosen DSM clustering algorithm, modular architectures for the product family can be compared.

Ideally, to obtain the optimal modular architecture for a given product variant, the components of the product variant would be clustered into modules without consideration of other product variants in the family. However, designing modules that are common to multiple product variants add restrictions to how the product variants are modularized. The minimum module clustering cost of a given product is obtained when the DSM of each product variant is clustered without consideration of commonality. Knowing this, the clustering cost of a sub-optimally clustered product variant, modularized as part of a product family, can be compared with the minimum possible clustering cost of that product variant. Comparing with the minimum clustering cost allows any proposed clustering of a product variant to be evaluated and quantified as a modularity score.

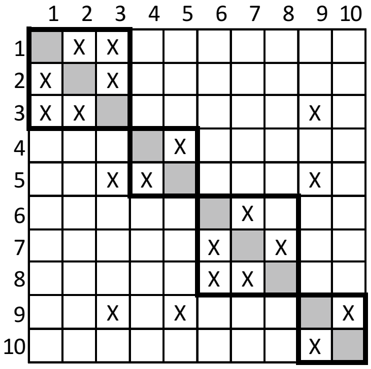
We use an example to demonstrate the calculation of a modularity score for a product variant in a family with a predefined platforming strategy. Figure 4 shows an example of a platforming strategy for a product family and the DSM for Product 1. To determine the modularity score of Product 1, the minimum clustering cost of the product is first calculated using a chosen DSM clustering algorithm. In this example, Thebeau's (2001) DSM algorithm is employed. The optimally clustered DSM for Product 1 is shown in Fig. 5. The optimal clustering of Product 1 consists of four modules ({1 2 3}, {4 5}, {6 7 8}, and {9 10}). This clustering has a cost of 71. Thus, the minimum DSM clustering cost of Product 1 is 71.

The component clustering restrictions imposed by the proposed platforming strategy are then enacted. With these restrictions in place, each of shared groups of the platforming strategy is individually clustered using Thebeau's algorithm. The optimal clustering of each of the shared groups that are common to Product 1 is given in Fig. 6. These shared groups are then combined back into one DSM to form the modular architecture of Product 1. Figure 7 shows the resultant modular architecture for Product 1. The regrouped DSM has a clustering cost of 140. This cost is compared to the optimal DSM cost (71) to calculate a ratio as the modularity score for this family modular architecture. The equation to calculate the modularity score is



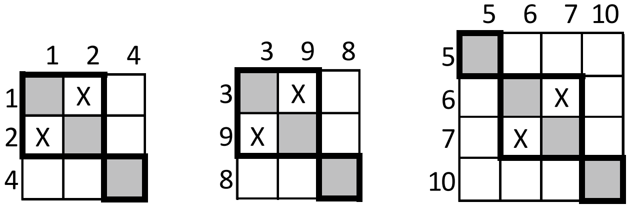
	1	2	3	4	5	6	7	8	9	10
1	▨	X	X							
2	X	▨	X							
3	X	X	▨							
4			X	▨	X					
5			X	X	▨				X	
6				X	X	▨	X			
7							▨			
8						X	X	▨		
9			X				X	X	▨	X
10									X	▨

Fig. 4 Platforming strategy and DSM of Product 1 in an example product family



	1	2	3	4	5	6	7	8	9	10
1	▨	X	X							
2	X	▨	X							
3	X	X	▨						X	
4				▨	X					
5			X	X	▨				X	
6						▨	X			
7						X	▨	X		
8						X	X	▨		
9			X		X				▨	X
10									X	▨

Fig. 5 Optimally clustered DSM of Product 1

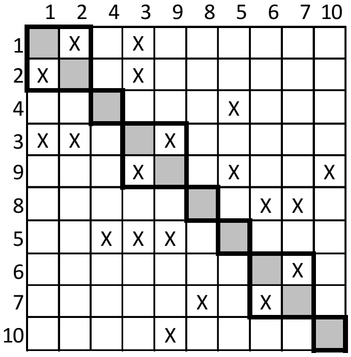


	1	2	4
1	▨	X	
2	X	▨	
4			▨

	3	9	8
3	▨	X	
9	X	▨	
8			▨

	5	6	7	10
5	▨			
6		▨	X	
7		X	▨	
10				▨

Fig. 6 Three shared groups of Product 1 with individual DSM clustering



	1	2	4	3	9	8	5	6	7	10
1	▨	X		X						
2	X	▨		X						
4			▨				X			
3	X	X		▨	X					
9				X	▨		X			X
8						▨		X	X	
5			X	X	X		▨			
6								▨	X	
7						X		X	▨	
10					X					▨

Fig. 7 DSM of the optimal clustering of Product 1 given the proposed family platforming strategy

$$\text{Modularity score for product } i \; MS_i = \left(\frac{\text{DSM clustering cost}_{\text{optimal},i}}{\text{DSM clustering cost}_{\text{proposed},i}} \right) \times 100 \tag{4}$$

$$\text{Example:} \quad \mathrm{MS}_1 = \frac{71}{140} \times 100 = 50.7$$

Using this technique, we can calculate a score for the modular clustering of each product variant in the family under the proposed platforming strategy. By weighting each of product variants, we determine a single modularity score for the product family architecture. Weighting each of product variants also allows more flexibility in controlling the design process, as product variants deemed to be more important to the success of the product family may be weighted more heavily than other products.

We assume Product 2 and Product 3 have individual modularity scores of 33.0 and 60.0 ($\mathrm{MS}_2 = 33.0$, $\mathrm{MS}_3 = 60.0$) and each of product variants is weighted equally ($w_1 = w_2 = w_3 = 1/3$). The modularity score for the product family is calculated as

$$\text{Modularity score for product family} = \mathrm{MS}_{\mathrm{PF}} = \sum_{i}^{N} w_i \cdot \mathrm{MS}_i \tag{5}$$

$$\text{Example:} \quad \mathrm{MS}_{\mathrm{PF}} = \frac{50.7 + 33.0 + 60.0}{3} = 47.9$$

The modularity score can then be used to compare different modular architectures identified from different platforming strategies. To obtain modular product family architectures with higher modularity scores, commonality has to be decreased. Thus, various options for architecting the product family could be plotted as a Pareto front of maximized commonality and modularity score.

3.5.3 Other metrics for product family design

We develop the procedure to generate Pareto front of maximum commonality and strategic modularity. The Pareto front provides designers multiple component sharing options to build a product family. The other design metrics of the product family, such as cost (Fujita and Yoshida 2004), technical performance (Fellini et al. 2004), complexity (Kim et al. 2016), sustainability (Kim and Moon 2017), and adaptability (Engel et al. 2017; Engel and Reich 2015) for industrial products, or visual appeal and ergonomics (Otto and Wood 2001) for consumer products, could then be evaluated for each of the component sharing options. These options could be compared and further modified, and designers can utilize area-specific expertise in this process. Importantly, the modular architecture of the product family derived by DSM to compute modularity score in Sect. 3.5.2 does not need to be the final product family modular architecture. In addition, designers also can substitute the other design metric for strategic modularity to construct customized Pareto front for their specific product family platform selection problem. For example, designers can build a Pareto front of maximum commonality and a technical performance in a similar manner to derive multiple component sharing options for further evaluation.

Of note, the cost of a product family may be considered in our method. Braha (2002) addressed that sharing components or modules between product variants in a family requires additional development and coordination costs. The costs may be represented by the weight *w* in modularity score calculation. The modularity strategy for individual product variant that leads to higher cost will have a smaller weight value. The mathematical relationship between the cost and the weight may be derived by empirical studies in future research.

The minimum number of building blocks of a product family also may be calculated from our method. In many cases, modularity may be used in a product family to organize the product variants into building block modules, which facilitate assembly and configurability. If the goal of designing the product family is to minimize the number of modules required to assemble the family, the minimum number of building block modules could be calculated by utilizing a DSM that represents the physical connections between components. This connectivity DSM may be the same as that used to complete the primary clustering of modules, or could be included in addition. After organizing the product family into modules based on the clustering DSM, two or more of these modules may be combined to form building blocks. Within each of shared groups in the product family, the identified modules are subsequently analyzed using the connectivity DSM to determine if the components of each module possess connections with components in other modules which allow them to be combined into a building block. By combining all modules in each shared group that is interconnected, the minimum number of building block modules needed to assemble the product family can be calculated. Here, the proposed minimum number of building blocks is only a reference for designers. There are many methodologies that suggest a balance between high-level and low-level granularity (AlGeddawy and ElMaraghy 2013; Chiriac et al. 2011; Koh et al. 2015), which are valuable for designers to build an appropriate product family from the component sharing options derived by our method.

4 A case study of impact driver and electric drill family design

To demonstrate the usefulness of our method, we apply the method to the product family platform selection of high-end and low-end impact drivers and electric drills. The product family includes a low-end electric drill, a low-end impact driver, a high-end brushless electric drill, and a high-end brushless impact driver. The list of product and components in the product family are given in Tables 2 and 3. The PCM for the product family is provided in “Appendix A”.

The interactions between the components in each of product variants are represented in a single binary DSM. DSM clustering and the computing of clustering costs is completed using the clustering algorithm presented by Thebeau (2001). Each of the product variants’ DSM is created. The DSM for the product family and each of individual product variants is given in Fig. 8. The optimally clustered DSMs for each of product variants are included in “Appendix B”.

Using the proposed method with the PCM and DSM inputs for the product family, we generate a Pareto front of platforming architectures with maximum commonality and modularity score. Each of the points on the Pareto front corresponds to a number of building blocks needed to construct the family. The Pareto front is identified, starting from the maximum commonality architecture. To search for Pareto optimal architectures, components of high commonality are iteratively moved into shared groups of lower levels of commonality that contain components with which they possess DSM interactions. Architectures that demonstrated higher modularity score are kept, while those that do not are discarded. The DSM in Fig. 8 is used as the connectivity DSM when calculating the number of building blocks needed for assembly. Figure 9 shows the Pareto fronts of platforming architecture options produced by our method. Unlike existing design optimization approaches, our method provides multiple component sharing options for designers’ further evaluation.

From the Pareto front in Fig. 9, we highlight three architectures for further analysis. The architectures with optimal modularity and maximum commonality are chosen to demonstrate the extremes of the Pareto front. An architecture balanced between these two metrics is chosen to exemplify an alternative platforming strategy that might be chosen by designers from the Pareto front. The assessment metrics for each of these architectures are presented in Table 4.

Tables 5, 6 and 7 present the modular architectures with maximum commonality, optimal modularity, and balanced commonality and strategic modularity, respectively. Figures 10, 11, and 12 demonstrate the component sharing within the family. In these figures, the modules are shown to

Table 2 Product list of a power tool product family

Product no.	Product description
1	Low-end electric drill
2	Low-end impact driver
3	High-end electric drill
4	High-end impact driver

Table 3 Component list of the power tool product family

Component no.	Component description	Component no.	Component description
C1	Clamshell	C21	VSR switch HE
C2	Armature 1	C22	Heat sink
C3	Armature 2	C23	Electronics board
C4	Field 1	C24	Belt clip
C5	Field 2	C25	Bit clip
C6	Motor brushes	C26	Drill light
C7	Brush holders	C27	Impact driver light
C8	Commutator	C28	Chuck HE
C9	Front bearing	C29	Chuck LE
C10	Pinion gear	C30	Transmission LE
C11	Rear bearing	C31	Transmission HE
C12	Motor fan	C32	Impact mech LE
C13	Motor ESC	C33	Impact mech HE
C14	Rotor magnet 1	C34	Anvil LE
C15	Rotor magnet 2	C35	Anvil HE
C16	Stator 1	C36	Nose cone
C17	Stator 2	C37	Battery terminal
C18	Trigger	C38	20V battery
C19	Fwd/rev switch	C39	Grip LE
C20	VSR switch LE	C40	Grip HE

exhibit the components in each module. The modules on the figures do not necessarily depict the specific detail design of each module. Components in each module would ideally be designed for high interaction between components within the module and low interaction between components outside of the module. Designers could modify each of the modules based on their own preference or other design considerations from the component sharing options provided by our method.

In the architecture with maximum commonality, every component that can be shared between product variants in the family is shared, resulting in a high commonality (TCCI) of 0.524. However, this commonality adds restrictions to the component clustering that result in higher clustering costs and a modularity score of only 58.0. This platforming architecture also results in a large number of small modules that cannot be combined into larger building blocks.

The architecture with the optimal modularity score has a modularity score of 100. In this architecture, each of the product variants is individually clustered to produce the minimum possible clustering cost. While this clustering provides the ideal clustering of each product variant, few of the components and modules may be shared between product variants in the family resulting in a commonality (TCCI) of 0.183.

The third highlighted platforming architecture provides a compromise of commonality and modularity score. This architecture has a commonality (TCCI) of 0.354 and a modularity score of 88.9. This balanced platforming architecture also requires 10 building blocks to construct in minimum, much less than that required by the platforming architecture with maximum commonality.

This case study and the logic behind this paper show a tradeoff between commonality and the quality of the modular architecture in product family design. This technique can informally be utilized, even without the implementation of DSM or other algorithmic approaches. Modules in a product family can be identified by first deciding the commonality of performance driving components and then evaluating the optimal tradeoff between the commonality of auxiliary components and quality of the modular architecture, keeping in mind that commonality can be sacrificed to provide a better strategic modular architecture.

5 Conclusion and future work

We introduce a method for product family platform selection with a balance of commonality and strategic modularity. We divide the components of a product family into groups based on the set of products between which they may be shared. Each of groups is then clustered into modules using DSM clustering and combined to form the complete modular architectures of the product family. Further alternative modular architectures and platforming strategies are considered by decreasing the amount of sharing in the product family to move common components to groups of less commonality which contain components with which they are highly

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	X																																						X	X
2		X						X	X	X	X																													
3			X					X	X	X	X																													
4		X		X																																				
5			X		X																																			
6						X	X	X																																
7	X					X	X																																	
8		X	X					X																																
9	X	X	X						X					X	X																									
10		X	X							X				X	X															X	X	X	X							
11	X	X	X								X			X	X																									
12		X	X									X		X	X																									
13													X			X	X																							
14									X	X	X			X		X																								
15									X	X	X				X		X																							
16	X												X	X		X																								
17	X												X		X		X																							
18																		X		X	X																			
19																			X	X	X																			
20																		X	X	X			X														X			
21																		X	X		X		X																	
22																						X	X																	
23													X							X	X	X	X														X			
24	X																							X																
25	X																								X															
26	X																						X			X														
27																				X			X				X									X				
28																												X												
29																													X											
30																													X	X										
31																												X			X									
32																																X		X		X				
33																																	X		X	X				
34																																		X						
35																																			X					
36	X																										X					X	X	X	X	X				
37	X																			X			X														X			
38																																						X		
39	X																																						X	
40	X																																							X

Fig. 8 DSM of the power tool product family

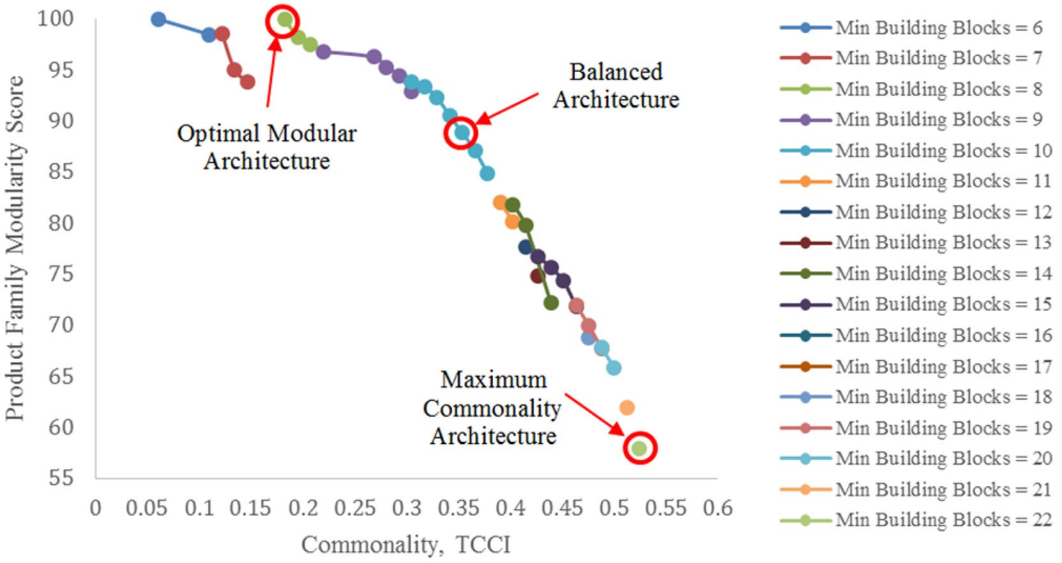


Fig. 9 Pareto front of component sharing options with maximized commonality and strategic modularity for the power tool family

Table 4 Assessment metrics for the three highlighted platforming architectures

Architecture	Commonality TCCI	Modularity score	Required number of building blocks
Max commonality	0.524	58.0	22
Max modularity score	0.183	100	8
Balanced	0.354	88.9	10

Table 5 Modular architecture of the product family design with maximum commonality

Number of products shared	Building block	Module	Components
4	B1	M1	C1 C9 C11 C24 C37
4	B2	M2	C10
4	B3	M3	C12
4	B4	M4	C18
4	B5	M5	C19
4	B6	M6	C38
2	B7	M7	C6 C7 C8
2	B8	M8	C20
2	B9	M9	C39
2	B10	M10	C13 C21 C22 C23
2	B11	M11	C40
2	B12	M12	C25
2	B13	M13	C27 C36
1	B14	M14	C2 C4
1	B15	M15	C29 C30
1	B16	M16	C3 C5
1	B17	M17	C32 C34
1	B18	M18	C14 C16
1	B19	M19	C26
1	B20	M20	C28 C31
1	B21	M21	C15 C17
1	B22	M22	C33 C35

Table 6 Modular architecture of the product family design with maximum modularity score

Number of products shared	Building block	Module	Components
4	B1	M1	C38
2	B2	M2	C6 C7
2	B3	M3	C18 C19 C20 C37
2	B4	M4	C18 C19 C21 C22 C23 C37
1	B5	M5	C1 C24 C25 C39
		M6	C2 C4 C8 C9 C10 C11 C12
		M7	C29 C30
1	B6	M8	C1 C24 C39
		M9	C3 C5 C8 C9 C10 C11 C12
		M10	C27 C32 C35 C36
1	B7	M11	C1 C24 C25 C26 C40
		M12	C9 C10 C11 C12 C13 C14 C16
		M13	C28 C31
1	B8	M14	C1 C24 C40
		M15	C9 C10 C11 C12 C13 C15 C17
		M16	C27 C33 C35 C36

Table 7 Modular architecture of the product family design with balanced commonality and strategic modularity

Number of products shared	Building block	Module	Components
4	B1	M1	C12
4	B2	M2	C38
2	B3	M3	C1 C11 C24 C39
		M4	C6 C7 C8
		M5	C18 C19 C20 C37
2	B4	M6	C1 C11 C24 C40
		M7	C18 C19 C21
		M8	C13 C22 C23 C37
2	B5	M9	C25
1	B6	M10	C2 C4 C9 C10
		M11	C29 C30
1	B7	M12	C3 C5 C9 C10
		M13	C27 C32 C35 C36
1	B8	M14	C9 C10 C14 C16
		M15	C28 C31
1	B9	M16	C26
1	B10	M17	C9 C10 C15 C17
		M18	C27 C33 C35 C36

coupled. To evaluate the component sharing options, we score the proposed product family platforming architectures based on commonality and the quality of the modular architecture. Taking advantage of algorithmically based DSM clustering, we determine a Pareto front of maximum commonality and strategic modularity. The component sharing options along the Pareto front could be evaluated, compared, and further modified based on designers' preference and/or other product family metrics (e.g., cost, technical performance, complexity, sustainability, adaptability, visual appeal, and ergonomics). We use the product family design of high-end and low-end impact drivers and electric drills as

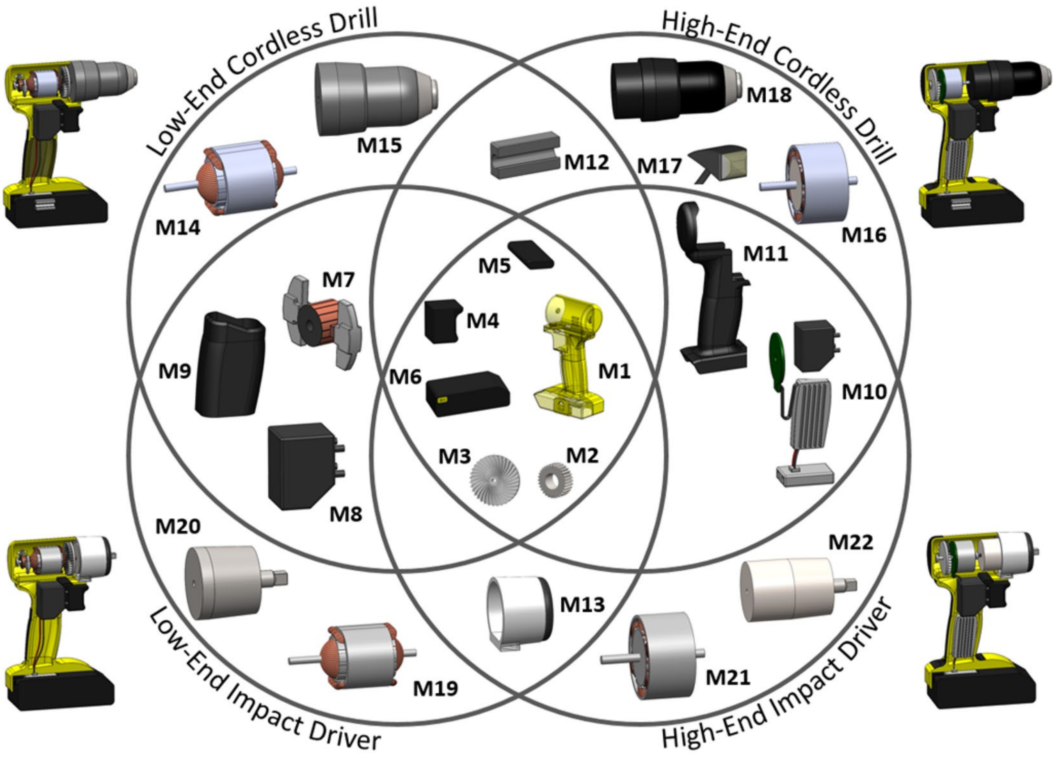


Fig. 10 Component sharing in the architecture with maximum commonality

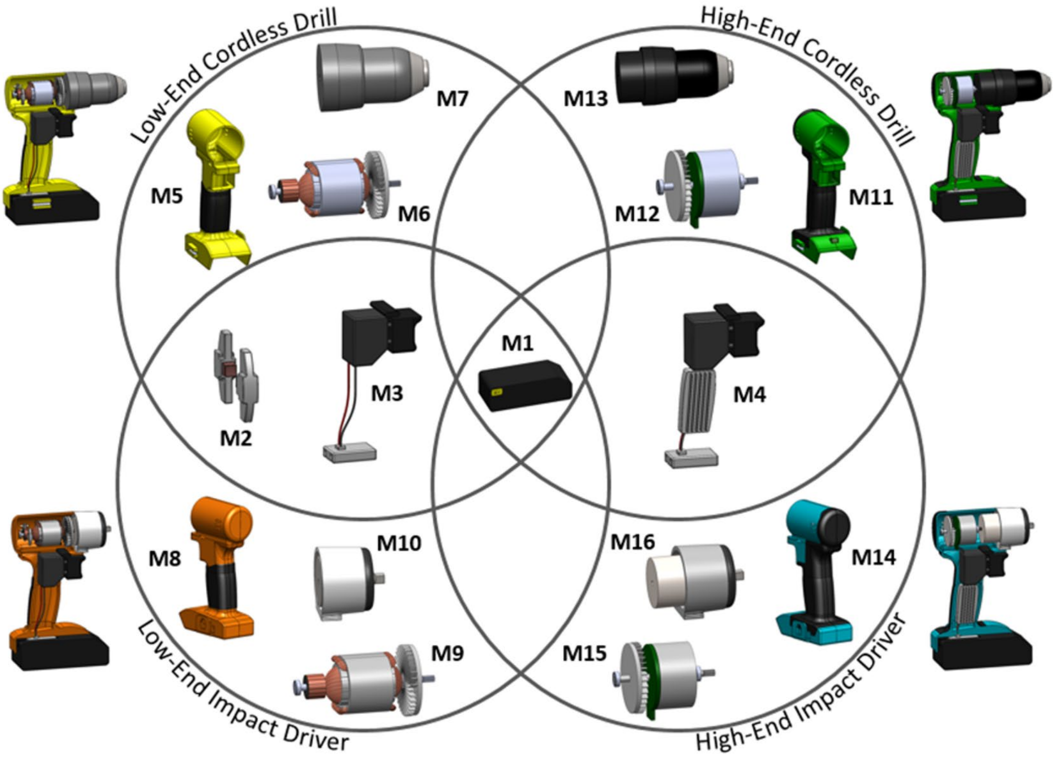


Fig. 11 Component sharing in the architecture with optimal modularity

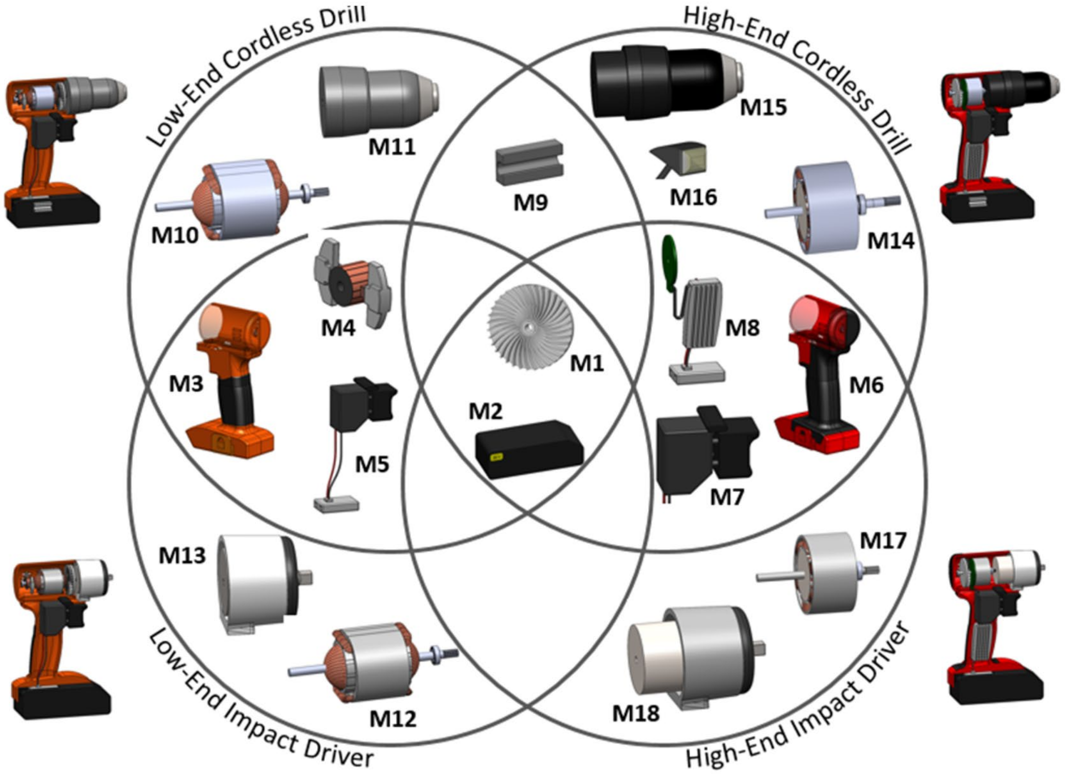


Fig. 12 Component sharing in the architecture with balanced commonality and strategic modularity

a case study to demonstrate our method. Three platforming strategies that lie on the Pareto front are highlighted.

Our method is broadly applicable and designers can apply our method not only in family design of industrial products but also use the method in consumer product design where some product metrics cannot be formulized. The method is compatible with any commonality indices and component clustering methods. Designer also can substitute the other product family metric (e.g., cost or technical performance) for strategic modularity to construct customized Pareto front. Moreover, our method helps designers address the uncertainty in the design process by providing multiple component sharing options, which is different from existing design optimization approaches. Designers can utilize area-specific expertise in the following decision-making process.

Our work has limitations that provide opportunities for future research. As the size of the product family increases, the computational time greatly increases. For a significantly large product family, the product family may need to be broken up and analyzed separately, determining the clustering of different sections of the product family independently. To reduce the computation time, advanced search algorithms may be developed in future research to select component sharing alternatives more strategically. In addition, researcher may improve the grading policy of strategic modularity presented in Sect. 3.5.2. In future research, a new modularity score that considers the cost or the technical performance of a product family may be developed.

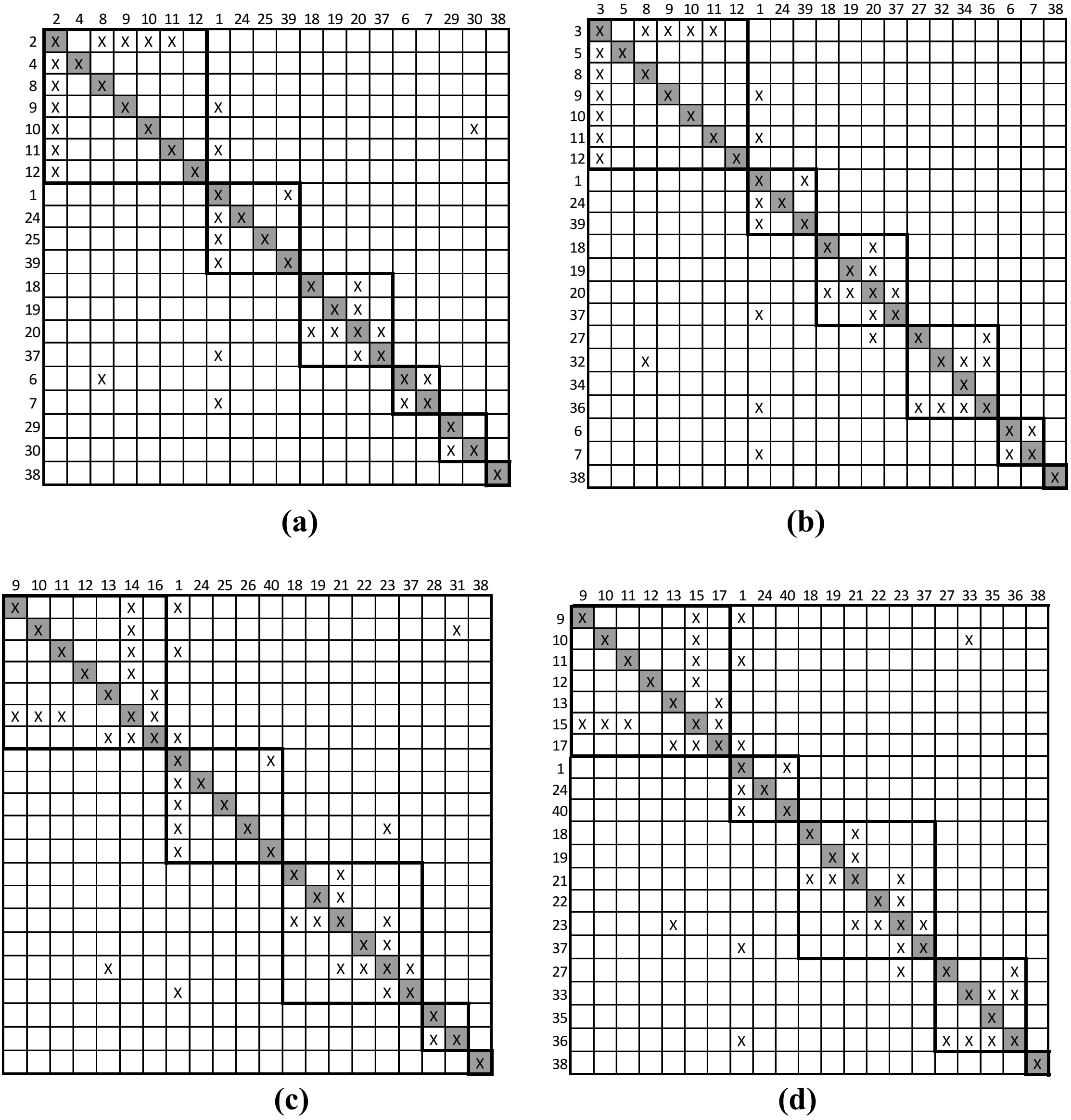
Acknowledgements This material is based in part on work supported by the National Science Foundation under Award number CMMI-1200256. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Table 8 The product–component matrix (PCM) of power tool product family

No.		Low-end cordless drill	Low-end impact driver	High-end cordless drill	High-end impact driver
1	Clamshell	1	1	1	1
2	Armature 1	1	0	0	0
3	Armature 2	0	1	0	0
4	Field 1	1	0	0	0
5	Field 2	0	1	0	0
6	Motor brushes	1	1	0	0
7	Brush holders	1	1	0	0
8	Commutator	1	1	0	0
9	Front bearing	1	1	1	1
10	Pinion gear	1	1	1	1
11	Rear bearing	1	1	1	1
12	Motor fan	1	1	1	1
13	ESC	0	0	1	1
14	Rotor magnet 1	0	0	1	0
15	Rotor magnet 2	0	0	0	1
16	Stator 1	0	0	1	0
17	Stator 2	0	0	0	1
18	Trigger	1	1	1	1
19	Fwd/rev switch	1	1	1	1
20	VSR switch LE	1	1	0	0
21	VSR switch HE	0	0	1	1
22	Heat sink	0	0	1	1
23	Electronics board	0	0	1	1
24	Belt clip	1	1	1	1
25	Bit clip	1	0	1	0
26	Drill light	0	0	1	0
27	Impact driver light	0	1	0	1
28	Chuck HE	0	0	1	0
29	Chuck LE	1	0	0	0
30	Transmission 1	1	0	0	0
31	Transmission 2	0	0	1	0
32	Impact mech LE	0	1	0	0
33	Impact mech HE	0	0	0	1
34	Anvil LE	0	1	0	0
35	Anvil HE	0	0	0	1
36	Nose cone	0	1	0	1
37	Battery connector	1	1	1	1
38	20V battery	1	1	1	1
39	Grip LE	1	1	0	0
40	Grip HE	0	0	1	1

Appendix A: The product–component matrix (PCM) of the power tool product family

Table 8 displays the transposed PCM for the power tool case study referenced in Sect. 4. It contains the list of components that constitute each of the product variants in the case study.



	2	4	8	9	10	11	12	1	24	25	39	18	19	20	37	6	7	29	30	38
2	X		X	X	X	X														
4	X	X																		
8	X		X																	
9	X			X				X												
10	X				X														X	
11	X					X		X												
12	X						X													
1								X			X									
24								X	X											
25								X		X										
39								X			X									
18												X		X						
19													X	X						
20												X	X	X	X					
37								X						X	X					
6			X													X	X			
7								X								X	X			
29																		X		
30																		X	X	
38																				X

(a)

	3	5	8	9	10	11	12	1	24	39	18	19	20	37	27	32	34	36	6	7	38
3	X		X	X	X	X															
5	X	X																			
8	X		X																		
9	X			X				X													
10	X				X																
11	X					X		X													
12	X						X														
1								X		X											
24								X	X												
39								X		X											
18											X		X								
19												X	X								
20											X	X	X	X							
37								X					X	X							
27													X		X			X			
32			X													X	X	X			
34																	X				
36								X							X	X	X	X			
6																			X	X	
7								X											X	X	
38																					X

(b)

	9	10	11	12	13	14	16	1	24	25	26	40	18	19	21	22	23	37	28	31	38
9	X					X		X													
10		X				X														X	
11			X			X		X													
12				X		X															
13					X		X														
14	X	X	X			X	X														
16					X	X	X	X													
1								X				X									
24								X	X												
25								X		X											
26								X			X						X				
40								X				X									
18													X		X						
19														X	X						
21													X	X	X		X				
22																X	X				
23					X										X	X	X	X			
37								X									X	X			
28																			X		
31																			X	X	
38																					X

(c)

	9	10	11	12	13	15	17	1	24	40	18	19	21	22	23	37	27	33	35	36	38
9	X					X		X													
10		X				X												X			
11			X			X		X													
12				X		X															
13					X		X														
15	X	X	X			X	X														
17					X	X	X	X													
1								X		X											
24								X	X												
40								X		X											
18											X		X								
19												X	X								
21											X	X	X		X						
22														X	X						
23					X								X	X	X	X					
37								X							X	X					
27															X		X			X	
33																		X	X	X	
35																			X		
36								X									X	X	X	X	
38																					X

(d)

Fig. 13 Optimally clustered DSM matrices for the **a** low-end cordless drill, **b** low-end impact driver, **c** high-end cordless drill, and **d** high-end impact driver

Appendix B: Optimal component clusters of product variants in the power tool product family

Figure 13 shows the optimal clusters of product variants in the power tool case study referenced in Sect. 4. Each modular architecture alternative was compared with the clustering costs of this optimal clustering to calculate the modularity scores of the product variants and the product family.

References

AlGeddawy T, ElMaraghy H (2013) Optimum granularity level of modular product design architecture. CIRP Ann 62:151–154. https://doi.org/10.1016/j.cirp.2013.03.118

Bell ET (1934) Exponential numbers. Am Math Mon 41:411–419. https://doi.org/10.2307/2300300

Borjesson F, Hölttä-Otto K (2014) A module generation algorithm for product architecture based on component interactions and strategic drivers. Res Eng Des 25:31–51. https://doi.org/10.1007/s00163-013-0164-2

Braha D (2002) Partitioning tasks to product development teams. ASME DETC 2002:333–344 https://doi.org/10.1115/DETC2002/DTM-34031

Chiriac N, Hölttä-Otto K, Lysy D, Suk Suh E (2011) Level of modularity and different levels of system granularity. J Mech Des 133:101007. https://doi.org/10.1115/1.4005069

Eichstetter M, Müller S, Zimmermann M (2015) Product family design with solution spaces. J Mech Des 137:121401. https://doi.org/10.1115/1.4031637

Engel A, Reich Y (2015) Advancing architecture options theory: six industrial case studies. Syst Eng 18:396–414. https://doi.org/10.1002/sys.21312

Engel A, Browning TR, Reich Y (2017) Designing products for adaptability: insights from four industrial cases. Decis Sci 48:875–917. https://doi.org/10.1111/deci.12254

Eppinger SD, Browning TR (2012) Design structure matrix methods and applications. MIT press, Cambridge

Erens F, Verhulst K (1997) Architectures for product families. Comput Ind 33:165–178. https://doi.org/10.1016/S0166-3615(97)00022-5

Erixon G (1998) Modular function deployment: a method for product modularisation. Royal Inst. of Technology, Department of Manufacturing Systems, Assembly Systems Division

Fellini R, Kokkolaras M, Papalambros P, Perez-Duarte A (2004) Platform selection under performance bounds in optimal design of product families. J Mech Des 127:524–535. https://doi.org/10.1115/1.1899176

Fujita K, Yoshida H (2004) Product variety optimization simultaneously designing module combination and module attributes. Concurr Eng 12:105–118. https://doi.org/10.1177/1063293X04044758

Gebhardt N, Bahns T, Krause D (2014) An example of visually supported design of modular product families. Procedia CIRP 21:75–80. https://doi.org/10.1016/j.procir.2014.03.162

Gonzalez-Zugasti JP, Otto KN, Baker JD (2000) A method for architecting product platforms. Res Eng Des 12:61–72. https://doi.org/10.1007/s001630050024

Harlou U (2006) Developing product families based on architectures. Department of Mechanical Engineering, Technical University of Denmark

Helmer R, Yassine A, Meier C (2010) Systematic module and interface definition using component design structure matrix. J Eng Des 21:647–675. https://doi.org/10.1080/09544820802563226

Holtta KMM, Salonen MP (2003) Comparing three different modularity methods. ASME DETC 2003:533–541 https://doi.org/10.1115/DETC2003/DTM-48649

Jiao J, Simpson TW, Siddique Z (2007) Product family design and platform-based product development: a state-of-the-art review. J Intell Manuf 18:5–29. https://doi.org/10.1007/s10845-007-0003-2

Jose A, Tollenaere M (2005) Modular and platform methods for product family design: literature analysis. J Intell Manuf 16:371–390. https://doi.org/10.1007/s10845-005-7030-7

Jung S, Simpson TW (2016) An integrated approach to product family redesign using commonality and variety metrics. Res Eng Des 27:391–412. https://doi.org/10.1007/s00163-016-0224-5

Kim S, Moon SK (2017) Sustainable platform identification for product family design. J Clean Prod 143:567–581. https://doi.org/10.1016/j.jclepro.2016.12.073

Kim G, Kwon Y, Suh ES, Ahn J (2016) Analysis of architectural complexity for product family and platform. J Mech Des 138:071401. https://doi.org/10.1115/1.4033504

Koh ECY, Förg A, Kreimeyer M, Lienkamp M (2015) Using engineering change forecast to prioritise component modularisation. Res Eng Des 26:337–353. https://doi.org/10.1007/s00163-015-0200-5

Kristjansson AH, Jensen T, Hildre H-P (2004) The term platform in the context of a product developing company. In: DS 32: Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik

Lei N, Yao X, Moon SK, Bi G (2016) An additive manufacturing process model for product family design. J Eng Des 27:751–767. https://doi.org/10.1080/09544828.2016.1228101

Liu Z, Wong YS, Lee KS (2011) A manufacturing-oriented approach for multi-platforming product family design with modified genetic algorithm. J Intell Manuf 22:891–907. https://doi.org/10.1007/s10845-009-0365-8

Ma J, Kim HM (2016) Product family architecture design with predictive, data-driven product family design method. Res Eng Des 27:5–21. https://doi.org/10.1007/s00163-015-0201-4

Meyer MH, Lehnerd AP (1997) The power of product platforms. Simon and Schuster, New York

Meyer MH, Utterback JM (1993) The product family and the dynamics of core capability. Sloan Manag Rev 34:29–47

Moon SK, Park KJ, Simpson TW (2014) Platform design variable identification for a product family using multi-objective particle swarm optimization. Res Eng Des 25:95–108. https://doi.org/10.1007/s00163-013-0166-0

Nelson IISA., Parkinson MB, Papalambros PY (2000) Multicriteria optimization in product platform design. J Mech Des 123:199–204. https://doi.org/10.1115/1.1355775

Östgren B (1994) Modularisation of the product gives effects in the entire production. The Royal Institute of Technology, Stockholm

Otto KN, Wood KL (2001) Product design: techniques in reverse engineering and new product development. Vol book, whole. Prentice-Hall, Inc., Upper Saddle River

Pirmoradi Z, Wang GG, Simpson TW (2014) A review of recent literature in product family design and platform-based product development. In: Simpson TW, Jiao J, Siddique Z, Hölttä-Otto K (eds) Advances in product family and product platform design: methods & applications. Springer New York, New York, pp 1–46. https://doi.org/10.1007/978-1-4614-7937-6_1

Robertson D, Ulrich K (1998) Planning for product platforms. Sloan Manag Rev 39:19–31

Simpson TW, Siddique Z, Jiao RJ (2006) Product platform and product family design: methods and applications. Springer Science & Business Media, New York

Simpson TW, Jiao J, Siddique Z, Hölttä-Otto K (2014) Advances in product family and product platform design. Springer, New York

Steward DV (1981) The design structure system: a method for managing the design of complex systems. IEEE Trans Eng Manag 28:71–74. https://doi.org/10.1109/TEM.1981.6448589

Stone RB, Wood KL, Crawford RH (2000) Using quantitative functional models to develop product architectures. Des Stud 21:239–260. https://doi.org/10.1016/S0142-694X(99)00008-3

Thebeau RE (2001) Knowledge management of system interfaces and interactions from product development processes. Massachusetts Institute of Technology, Cambridge

Thevenot HJ, Simpson TW (2006) Commonality indices for product family design: a detailed comparison. J Eng Des 17:99–119. https://doi.org/10.1080/09544820500275693

Ulrich K (1994) Fundamentals of product modularity. In: Dasu S, Eastman C (eds) Management of design: engineering and management perspectives. Springer Netherlands, Dordrecht, pp 219–231. https://doi.org/10.1007/978-94-011-1390-8_12

Wacker JG, Treleven M (1986) Component part standardization: an analysis of commonality sources and indices. J Oper Manag 6:219–244. https://doi.org/10.1016/0272-6963(86)90026-4

Wang D, Du G, Jiao RJ, Wu R, Yu J, Yang D (2016) A Stackelberg game theoretic model for optimizing product family architecting with supply chain consideration. Int J Prod Econ 172:1–18. https://doi.org/10.1016/j.ijpe.2015.11.001

Yassine A, Braha D (2003) Complex concurrent engineering and the design structure matrix method. Concurr Eng 11:165–176. https://doi.org/10.1177/106329303034503

Yu T-L, Yassine AA, Goldberg DE (2007) An information theoretic method for developing modular architectures using genetic algorithms. Res Eng Des 18:91–109. https://doi.org/10.1007/s00163-007-0030-1

Zamirowski EJ, Otto KN (1999) Identifying product family architecture modularity using function and variety heuristics. In: 11th International Conference on Design Theory and Methodology, ASME, Las Vegas

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.