European Journal of Operational Research 000 (2016) 1–15

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Decision Support

An efficient solution method to design the cost-minimizing platform portfolio

Maud Van den Broeke[a,b,c,*], Robert Boute[b,c], Brecht Cardoen[b,c], Behzad Samii[b,d]

[a] Management Department, IESEG School of Management, Lille, France
[b] Technology and Operations Management Area, Vlerick Business School, Gent, Belgium
[c] Research Center for Operations Management, KU Leuven, Leuven, Belgium
[d] Zaragoza Logistics Center, Zaragoza, Spain

ARTICLE INFO

Article history:
Received 11 July 2015
Accepted 3 October 2016
Available online xxx

Keywords:
OR in Research & Development
Platform design
Metaheuristic
Fathoming rules

ABSTRACT

To offer a wide product variety to customers in a cost-efficient way, companies have introduced platforms, defined as a base from which different products can be derived. We consider a product portfolio, consisting of a set of end products, where each product has a set of attributes (features), which can have different levels requested by the customers. We present a model to support companies in designing the cost-minimizing platform portfolio, consisting of a set of platforms, from which these products can be derived. Each platform has a set of technical design parameters, which can have different levels. The required parameter levels in the platform's design depend on the attribute needs of the products derived from the platform. Our model gives guidance to what extent the platforms should be under-designed, over-designed or the same with regard to the products (and its attribute levels) derived from them. The model quantifies the impact of these platform portfolio decisions on the relevant operational costs. Given the complexity of this problem for large-scale instances, we develop two fathoming rules to improve computational efficiency. These fathoming rules can be used in different solution algorithms. We illustrate their applicability in a branch-and-bound, simulated annealing and genetic algorithm. We demonstrate the value of our model and solution method with a practical case of a high-tech screen manufacturing company, that wants to design the cost-minimizing platform portfolio from which their product portfolio can be derived.

1. Introduction

In the quest to fulfill customers' demand for high product variety in a cost-efficient way, several companies have introduced platforms (Robertson & Ulrich, 1998). Meyer and Lehnerd (1997) define platforms as a 'set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently developed and produced'. Similar to the approach of Ben-Arieh, Easton, and Choubey (2009), we consider the use of multiple platforms, from which components are added or removed to customize and tailor the platforms to the individual products' requirements or attributes. We refer to the latter platform approach as '*customizable platforms*', which differs from the more common modular or scalable platform approaches in literature (Du, Jiao, & Chen, 2014).

The motivation for this research stems from a business case at a company specialized in the development and production of high-tech screens for use in healthcare. The medical screens are derived from (common) printed circuit boards ('pcb') by adding or removing (mechanical) components. The product portfolio offered to the market is driven by the competitor's offer and customers' requests, and is the result of decisions made by the sales & marketing department. Our high-tech screen manufacturing company struggles with the design of their platforms. Given their product portfolio, how should their platform portfolio optimally look like: how many platforms should be developed, and should these platforms be under-designed, over-designed or the same compared to the products derived from them?

This research is related to the product portfolio planning (PPP) literature, also referred to as product portfolio management or product family positioning (Jiao, Simpson, & Siddique, 2007a). PPP aims at selecting the 'right' mix of products and attribute levels to

* Corresponding author at: Management Department, IÉSEG School of Management, Lille, France.

E-mail addresses: m.vandenbroeke@ieseg.fr, maudvandenbroeke@hotmail.com (M. Van den Broeke), robert.boute@vlerick.com (R. Boute), brecht.cardoen@vlerick.com (B. Cardoen), behzad.samii@vlerick.com (B. Samii).

http://dx.doi.org/10.1016/j.ejor.2016.10.003

offer in the marketplace (Jiao & Zhang, 2005b). This differs from traditional product line design as it does not only optimize the mix of products, but also their configuration in terms of attributes. PPP typically involves two stages: (1) generating the product portfolio and (2) evaluating and selecting the best portfolio (Jiao & Zhang, 2005a). Traditionally, PPP was mainly tackled from a marketing perspective with a focus on offering a portfolio that maximized the margin between the customer-perceived utility and the price of a product, known as the consumer surplus (Kaul & Rao, 1995). Focus was on maximizing consumer surplus, expected utility (Krishnan & Gupta, 2001) and market share (Kohli & Krishnamurti, 1989), which are driven by customer-preferences, product functionalities, price, the presence of competitors, and the probability of a customer choosing a certain product (Goswami, Pratap, & Kumar, 2016). Another way to select the best product portfolio is to trade-off the expected return with the variance in return (risk) for a certain portfolio (Crama & Schyns, 2003). Given the close connection of the product variety and the firm's internal operational and design complexity and costs, recent literature emphasizes the importance of aligning engineering and marketing when making product portfolio decisions. Stone, Kurtadikar, Villanueva, and Arnold (2008) study a platform design for the product portfolio based on the knowledge of customer's needs: the core needs of the customer form the common platform structure, and distinctive needs form the differentiating modules. Likewise, the recent stream of PPP literature focuses on simultaneously leveraging customer and engineering concerns (denoted as the 'shared-surplus') when making product portfolio decisions (Jiao & Zhang, 2005b; Sadeghi, Alem-Tabriz, & Zandieh, 2011). These shared-surplus models consider customer preferences and utility, pricing, market share, probabilities of customers choosing a certain product, but also platform-based product costing. Müller and Haase (2016) comment that when the customer surplus is deterministic, the shared-surplus model changes into a produced surplus maximization model.

Analogous to the PPP literature, we consider a product portfolio as a set of products offered to the market, which are selected from a set of product profiles. Each product profile is a bundle of *functional attributes*, sometimes also referred to as features (e.g., the quality and color of the medical screen), with specific attribute *levels* (e.g., a low vs. high quality and gray vs. colored medical screen) desired by the customers (Albritton & McMullen, 2007; Kohli & Krishnamurti, 1989).

While PPP research supports the selection of the right mix of products and attribute levels to offer in the target market (i.e., determining the product portfolio) (Jiao & Zhang, 2005b), in this article we look for the right mix of platforms (i.e., determining the platform portfolio) from which the products in the product portfolio can be derived. These platforms are configured from a bundle of *technical design parameters* (e.g., the pixel pitch and power consumption of the pcb) with specific *levels* (e.g., a 50 or 70 watts, 0.1700 or 0.1575 millimeter pcb). Whereas mapping the customer needs to functional requirements is essential when identifying the product portfolio, translating these functional requirements into design parameters is crucial to identify the product platforms (Jiao et al., 2007a).

Products can be derived from its *matching* platform, which means that the platform has design parameter levels that exactly respond to the product's attribute level needs (Jiao et al., 2007a). Products can also be derived from platforms that are either *under-* or *over-designed* relative to a product's attribute levels. This respectively refers to a platform with a performance level that is lower or higher compared to the product's desired performance level (Krishnan & Gupta, 2001). The difference in performance level between the platform and product is denoted as the *'performance gap'*, which is the difference between the design parameter levels required by the product and the ones included in the platform from which that particular product is derived. The larger the performance gap, the less aligned a platform is with a product, and the more customization is required to derive the product from the platform (Fujita & Yoshida, 2004).

Our article contributes to the literature by designing the cost-minimizing platform portfolio, with the costs expressed in terms of the performance gap between platform and product. Van den Broeke, Boute, and Samii (2015) categorize the supply chain costs related to product-platform decisions (the costs related to its development, purchasing of materials and inventory requirements) into costs linked to the platforms and costs to customize the platforms into the final products. We use the latter model as the basis of our cost quantification, but in our article we express these operational costs in terms of the design parameter levels of the platform and those needed to support the product's attribute levels, e.g. the purchasing costs of a platform depend on the costs of the platform's design parameter levels. Also the customization costs to adapt a product from a platform depend on the performance gap between the platform and product. As such a company can opt to only develop one common platform for all products, leading to low platform costs but high customization costs; while the opposite is true when developing matching platforms for each product (leading to high platform costs but low customization costs).

By expressing the platforms in function of its design parameters and the products in function of its attributes, the number of product-platform combination scenarios increases rapidly for practical instances. This increases the computational complexity to find the cost-optimal solution excessively, making this problem NP-hard. Therefore we provide an efficient solution method to solve our problem.

In the literature, there are several studies that solve related problems using local search algorithms, such as (1) the steepest descent or best improvement algorithms, (2) the simulated annealing and tabu search algorithms, or (3) genetic algorithms (Vaessens, Aarts, & Lenstra, 1998). Simpson, Siddique, and Jiao (2006, p. 152) provide an overview of the different optimization algorithms that are used for various research questions in the platform and product design literature.

Simulated annealing (SA) has been used to determine the extent of scalable platforms (Seepersad, Hernandez, & Allen, 2000), to determine the optimal product portfolio (Crama & Schyns, 2003; Sadeghi et al., 2011) and to develop a modular product family so that the sum of similarities between components is maximized (Wang, Zhang, & Zhao, 2005). To determine the best set of modules, Agard and Penz, (2009) and Agard, Cheung, and Da Cunha (2006) respectively use SA supplemented with a clustering approach and *Genetic algorithms (GA)*. GA is another popular solution method in platform and product design literature (Balakrishnan & Jacob, 1996). For example, Yang et al. (2015) and Nepal, Monplaisir, and Famuyiwa (2012) use GA to jointly configure the product family and the supply chain, Jiao, Zhang, and Wang (2007b) use GA for a product portfolio planning problem, and Khajavirad, Michalek, and Simpson (2009) apply a two-level GA to select and design platforms and variants. Notice that although GA is a more complex heuristic, it does not always lead to better results than SA (Sadeghi et al., 2011). Another method that can be applied is *Branch-and-Bound* (B&B), which finds the optimal solution without having to go through complete enumeration (Demeulemeester & Herroelen, 2002). Fujita and Yoshida (2004) simultaneously optimize the module combination design and module attributes selection, hybridizing a B&B, GA and a successive quadratic programming method. Due to the complexity of platform design problems, the use of B&B algorithms is less common in the platform literature, as a pure B&B algorithm (without fathoming rules) typically results in long computation times.

Please cite this article as: M. Van den Broeke et al., An efficient solution method to design the cost-minimizing platform portfolio, European Journal of Operational Research (2016), http://dx.doi.org/10.1016/j.ejor.2016.10.003