A Stackelberg game theoretic model for optimizing product family architecting with supply chain consideration

Wang, D., Du, G., Jiao, R.J., Wu, R., Yu, J. and Yand, D.

NOTICE: this is the authors' version of a work that was accepted for publication in International Journal of Production Economics. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in International Journal of Production Economics, 172 1-18 0925-5273.

International Journal of Production Economics is available online at:

https://dx.doi.org/10.1016/j.ijpe.2015.11.001

A STACKELBERG GAME THEORETIC MODEL FOR OPTIMIZING PRODUCT FAMILY ARCHITECTING WITH SUPPLY CHAIN CONSIDERATION[+]

Danping Wang[1], Gang Du[*, 1], Roger J. Jiao[2], Ray Wu[3], Jianping Yu[1], Dong Yang[4]

Abstract: *Planning of an optimal product family architecture (PFA) plays a critical role in defining an organization's product platforms for product variant configuration while leveraging commonality and variety. The focus of PFA planning has been traditionally limited to the product design stage, yet with limited consideration of the downstream supply chain-related issues. Decisions of supply chain configuration have a profound impact on not only the end cost of product family fulfillment, but also how to design the architecture of module configuration within a product family. It is imperative for product family architecting to be optimized in conjunction with supply chain configuration decisions. This paper formulates joint optimization of PFA planning and supply chain configuration as a Stackelberg game. A nonlinear, mixed integer bilevel programming model is developed to deal with the leader-follower game decisions between product family architecting and supply chain configuration. The PFA decision making is represented as an upper-level optimization problem for optimal selection of the base modules and compound modules. A lower-level optimization problem copes with supply chain decisions in accordance with the upper-level decisions of product variant configuration. Consistent with the bilevel optimization model, a nested genetic algorithm is developed to derive near optimal solutions for PFA and the corresponding supply chain network. A case study of joint PFA and supply chain decisions for power transformers is reported to demonstrate the feasibility and potential of the proposed Stackelberg game theoretic joint optimization of PFA and supply chain decisions.*

Keywords: *Product family architecting, Supply chain configuration, Stackelberg game, Bilevel optimization, Nested genetic algorithm.*

[+] *This research is partially supported by the National Natural Science Foundation of China under Project Number 71371132.*

[*] *Corresponding author, 404 894 9633, tddg@tju.edu.cn*

[1] *College of Management and Economics, Tianjin University, Tianjin 300072, China*

[2] *The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA30332-0405, USA*

[3] *University of Westminster, UK*

[4] *School of Business and Management, Donghua University, Shanghai, 200051, China*

1. INTRODUCTION

Product family architecting aims at optimal planning of an underlying architecture of an organization's product platform based on commonality and planned variability, such that various product variants can be derived through module configuration. The challenge of product family decision making resides with how to reuse product components and structures throughout the product family while differentiating product variety with decreased costs and time (Jiao and Tseng, 1999). From the perspective of product design and development, a product architecture defines how the functional elements of a product are arranged into its physical units and how these units interact with one another (Ulrich and Eppinger, 1995). A product family architecture (PFA), on the other hand, deals with configuration of modules according to a given product architecture by distinguishing what are the common modules and structural design to be shared among product variants, and by optimizing product differentiation while leveraging upon the performance of the entire family (Jiao and Tseng, 2000).

The focus of PFA has been traditionally limited to the product design stage, yet with limited consideration of the downstream supply chain-related issues (Jiao et al., 2007). The fulfillment of product families is enacted through assembly-to-order production, which nowadays more and more involves globally distributed operations and manufacturers (Jiao et al., 2009), leading to such supply chain concerns as facility locations and node selection in a manufacturing supply chain network (Elmaraghy and Mahmoudi, 2009). Supply chain decisions affect not only the end cost of product family designed, but also the decision models of module configuration within a PFA (Huang et al., 2005). For example, product family configuration must take into account the implications and consequence of different outsourcing policies of PFA modules in the supply chain (Lamothe et al., 2006). The corresponding supply chain decisions to a PFA constitute a supply chain architecture (SCA) that addresses how to configure a supply chain for the product families, involving such configuration decisions as the selection of supply options at each echelon of the supply chain and the placement of inventory at each supply chain echelon (Truong and Azadivar, 2005). Therefore, optimal PFA planning is coupled with supply chain configuration and in turn joint decision making is deemed to be imperative (Shahzad and Hadj-Hamou, 2013).

Existing decision models for joint optimization of product families and supply chain configuration are originated from a basic assumption that the PFA and SCA decisions can be integrated into one single optimization problem by aggregating two different types of objectives into a single-level objective function through certain coordinated protocol, e.g., a weighted sum (Fujita et al., 2013). However, such an "all-in-one" approach neglects the complex tradeoffs underlying two different decision making problems and fails to reveal the inherent coupling of PFA and SCA (Jiao and Tseng, 2013). In practice, PFA decisions are mostly made by a company's designers, whereas SCA decisions are often attributed to many other companies in the supply chain that play their individual roles as suppliers, manufacturers, assembly plants, or distribute centers (DCs). Different priorities of decision making between the PFA and the SCA lead to many conflicting goals and constraints that must arrive at equilibrium solutions among diverse decision makers. Such joint optimization of product families and supply chain issues necessitates a non-cooperative game, which entails a leader-follower decision structure between the PFA and the SCA.

To handle the inherent interactions and hierarchical characteristics of joint decision making between two self-interested roles of PFA and SCA, we propose a Stackelberg game theoretic optimization model for coordinated product family architecting and supply chain configuration. Moreover, the existing literature on product family design mainly focuses on optimization of module configuration based on a given PFA, in which the modular architecture is already established, and thus contains a fixed number of decision variables. To the contrary, PFA planning is about how to design such a PFA by determining an optimal modular architecture. Due to the fact that the modular architecture is unknown before PFA planning concludes, PFA planning must deal with an uncertain number of decision variables.

The paper proceeds as follows. The state-of-the-art research is reviewed in Section 2. Section 3 defines the problem context of supply chain issues in PFA planning. The optimization problems of PFA planning and supply chain configuration are elaborated in Sections 4 and 5, respectively. Section 6 presents a bilevel optimization model that coincides with the game theoretic decision making process between a leader and a follower. The leader problem represents PFA planning for optimal combinations of product variants and the common modular structure. The follower problem deals with SCA decisions by observing the results of PFA planning and meanwhile possesses the autonomy in determining appropriate facility locations and operational variables for the suppliers, manufacturers, assembly plants and DCs. Consistent with the bilevel optimization model, a nested genetic algorithm is developed in Section 7 to derive near optimal solutions of PFA and SCA. Section 8 reports a case study of joint PFA and supply chain decisions for power transformer products, along with performance analysis of the proposed Stackelberg game theoretic joint optimization model.

2. RELATED WORK

2.1 Product Family Architecting

Any product family exhibits a certain form of an architecture that impacts on product performance, product upgrades, product variety, component standardization, manufacturability, and product change (Ulrich, 1995). Jiao and Tseng (2000) review the fundamental issues of PFA planning, including modularity and commonality, functional and technical variety, and multiple views of a PFA. Based on function analysis, Stone et al. (2004) propose a module assembly heuristic for product architecture conceptualization. Rodriguez and Ashaab (2005) develop a knowledge driven collaborative product development system to facilitate knowledge supply in product architecting. Zhu et al. (2010) apply rough sets and neural networks to predict performance of new product family configuration.

Towards the goal of optimal product variant configuration with limited resources, PFA planning calls for extensive applications of optimization techniques. For a balance of versatility and performance among product variants, D'Souza and Simpson (2003) use formal experiment design to identify important factors of product family design and develop a multi-objective genetic algorithm for product variant performance optimization. Fujita (2002) proposes a hybrid method that uses genetic algorithm, mixed integer programming and constrained nonlinear programming, respectively. Jiao and Zhang (2005) propose a 0-1 mathematical programming model for portfolio planning of a PFA that emphasizes customer-engineering interaction. Huang et al. (2005) propose to integrate product platform, process and supply chain decisions to minimize total supply costs and improve supply chain efficiency. Li and Huang (2009) consider PFA planning as a multi-objective optimization problem, considering product performance, product family penalty function, and degree of commonality in the objective function, leading to a multilayer evaluation method for product family analyses at different levels, i.e. product, module, component, and parameter. Cao et al. (2011) consider the life cycle costs in the optimization process, through mathematical programming models to reduce performance loss within the product range. Fujita et al. (2013) propose a mathematical model for simultaneous design of product families and the global supply chain configuration, through selecting of manufacturing sites, product assembly and distribution.

2.2 Product Family and Supply Chain Coordination

Lee et al. (2009) point out that companies need to integrate the supply chain and share the product information. Cheng (2011) shows that customization drives manufacturers using modular design model for managing their supply chain. Verdouw et al. (2010) observe that changes in product structures can influence the dynamics of supply chains, such as outsourcing and transferring production of more components to suppliers and combination of first-tier suppliers into mega suppliers. Doran (2003) also shows consequences of coupled product architecting and supply chain decisions, including reorganization of value creation activities where some former first-tier-suppliers become value-added second-tier supplier, suppliers becoming more powerful with an increased bargaining power because of the larger role as a full service supplier, and formation of more strategic alliances or partnerships between the OEMs and their suppliers.

More consensuses on focusing on product development and supply chain relationships at the product architecting stage have been reported in recent years. Pero et al. (2010) report case studies indicating that the performance of supply chain depends upon the matching between product development and supply chain decisions. Chiu and Okudan (2011) propose to combine design for assembly and supply chain configuration during the product development stage. Likewise Ulku and Schmidt et al. (2011) study how to the level of product modularity and supply chain configuration.

One of the recent focuses has been geared towards joint decision making of various stages in both product families and supply chains using an integrated approach. Shahzad and Hadj-Hamou (2013) adopt a general bill-of-product and a general supply-chain-structure to represent product families and supply chain issues. Zhang et al. (2008) apply linear optimization to design of product platforms concurrently with supply chain configurations. Du et al. (2013) establish a bilevel optimization model for product family configuration considering such supply chain issues as external module suppliers, internal manufacturing methods, and transportation in the supply chain configuration. Yang et al. (2015) formulate a bilevel joint optimization model for configuration of a product family and its supply chain, in which supply chain configuration determines the supply network and inventory policies for the suppliers, manufacturers, assemblers, DCs, and retailers.

2.3 Bilevel Programming

Bilevel programming basically instantiates a Stackelberg game to be a mathematical program that contains a sub-optimization problem in its constraints (Bracken et al., 1973). Since the upper level of the bilevel model contains the optimal solution or optimal value of the lower level, bilevel programs are generally non-smooth optimization problem, in which the feasible region of the upper level may not be connected. Even a linear bilevel programming is NP-hard (Jeroslow, 1985).

Stackelberg games entail a leader-follower decision structure (Von Stackelberg, 1952), which contains an upper-level optimization problem (referred to as a leader), along with one or more lower-level optimization problems (referred to as followers). The leader holds a powerful position in the hierarchical decision problem and the followers react rationally to the leader's decision (Gibbons, 1992). The leader-follower Stackelberg game formulation has been applied in a number of fields, such as design and maintenance (Hernandez et al., 2002), homogeneous product duopoly market research (Krishnendu, 2005), a two-stage supply chain with one manufacturer and one distributor (Qin, 2012).

Recent applications of bilevel programming in engineering design have indicated the potential of leader-follower decision making. For instance, Shabde and Hoo (2008) employ bilevel programming to find the optimal design of process control within a hierarchical decision making framework. Hernandez et al. (2002) formulate a leader–follower game-theoretic model for collaborative product design and maintenance management. Nonetheless, there is little application of bilevel Stackelberg games to PFA planning with supply chain consideration.

Common solution methods of bilevel programming method include the K times best for linear bilevel programming (Bard and Falk, 1982), using K-T conditions to replace the lower level and converting the problem to a single-level program (Fortuny and McCarl, 1981), using a dual-gap structure penalty function to convert the problem to a single-level problem (Anandalingam, 1990), and intelligent algorithms (Mathieu, 1994). Sakawa and Nishizaki (2012) review the interactive fuzzy methods for multilevel programming problems. Fliege and Vicente (2006) propose a multi-criteria method to bilevel programming. Colson et al. (2005) consider the approximation of non-linear bilevel mathematical programs by solvable programs of the same type. However, these methods not only are technically inefficient, but also lead to a paradox that the follower's decision power could dominate the leader's (Lai, 1996).

2.4 Genetic Algorithms

Genetic algorithms (GAs) have also been demonstrated potential for bilevel programming. Liu (1998) formulates a Stackelberg-Nash equilibrium with GAs for multilevel optimization. Niwa et al. (1998) adopt double strings in GAs for two-level 0-1 programming. Oduguwa and Roy (2002) propose a bilevel GA to encourage limited asymmetric cooperation between two players. Li and Wang (2008) incorporate a GA with Lemke algorithm. However, these efforts have to assume the follower's problem to be a convex quadratic programming problem so as to transform the bilevel model to a single-level problem using KKT conditions.

3. PFA PLANNING WITH SUPPLY CHAIN DECISIONS

The traditional task of product family configuration emphasizes the derivation of product variants by selecting optional modules based on a given PFA (Yang et al., 2015), whereas the focus of this paper is on PFA planning, that is, how to establish such a PFA. The critical issue of PFA planning is to identify the common modules and structures (i.e., product platform) to be shared among product variants by optimizing product differentiation while leveraging upon the performance of the entire product family. Fig. 1 illustrates the architecture of product family configuration, in which a product platform is exemplified by combinations of compound modules. Each compound module is composed by certain base modules and all the compound modules form the common modular structure to be shared by a product platform. The base modules are identified by clustering similar module instances that are to be fulfilled through the supply chain.

Assume that a product family serves a number of I market segments. The *i*-th market segment is associated with customer group $\mathrm{S_i}$ and characterized by annual demand $\mathrm{Q_i}$, where $\mathrm{i = 1,\ldots,I}$. Let $\mathrm{m_k^B}$ ($\mathrm{k = 1,\ldots,K}$) denote a base module and $\mathrm{m_{kl}^*}$ stand for the *l*-th module instance of base module $\mathrm{m_k^B}$ that contains a total number of $\mathrm{L_k}$ instances. Introduce a decision variable$\mathrm{R}$, to indicate the total number of compound modules, $\mathrm{m_r^C}$ ($\mathrm{r = 1,\ldots,R}$), to be identified through grouping base modules with similar module instances. The PFA planning involves another decision variable$\mathrm{J}$, which determines an appropriate number of product variants $\mathrm{P_j}$ ($\mathrm{j = 1,\ldots,J}$), that should be contained in a product family in terms of different configurations of compound modules. The base modules with different instances are either made in house by specific manufacturing plants or purchased from suppliers. Therefore, configuring a product variant is enacted through the choice of compound and base modules per se. As a result, critical PFA planning decisions is about optimization of the number of product variants, identification of compound modules, and choosing of base modules.

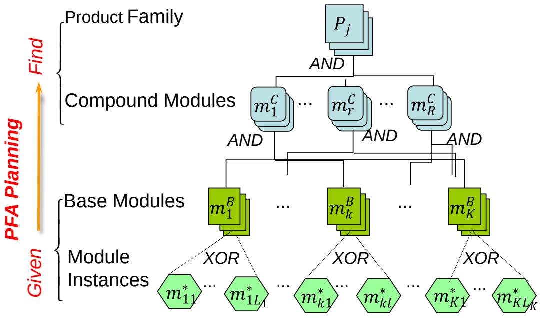

Fig. 1: PFA planning decisions

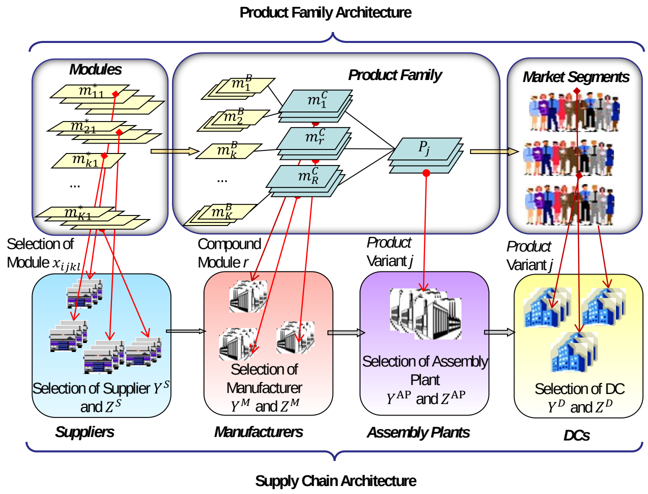

Fig. 2: Coupled decisions between a PFA and an SCA

Fig. 2 illustrates the fulfillment of a product family is achieved through a supply chain architecture (SCA), which addresses how to configure a supply chain in terms of selection of suppliers for the base modules, choice of manufacturing and assembly plants for the compound modules, and selection of distribute centers (DCs) for delivery of the final product variants, along with inventory and operations policies at each stage of the supply chain. It is assumed that all the suppliers, manufacturing and assembly plants, and DCs possess the right capabilities for order fulfilling in terms of quality, creditability, etc.

Consider a general case of assembly-to-order for product family configuration, in which the base modules are supposed to be purchased from the suppliers. Assume that each instance of a base module, m_{kl}^{*}, can be acquired from a number of S_{kl} possible suppliers, and thus supply chain decisions for m_{kl}^{*} are all about supplier selection. Likewise, assume each compound module, m_{r}^{C}, can be produced by a number of M_r possible manufacturers, for which selection of a best candidate manufacturer for m_{r}^{C} necessitates SCA decisions for compound modules. Similarly, fulfillment of a product variant configuration requires selection of a best candidate out of a number of AP_j possible assembly plants that are all capable of assembling product variantP_j. To deliver P_j to the customer, it is necessary select the right DC from a number of D_j candidate DCs.

4. Optimization of Product Family Architecting

The upper-level optimal PFA planning problem needs to determine the number of product variants, the number of compound modules, and the selection of base modules for every product variants. Let each design variable, $\vec{X}_j = (X_{j1},\ldots,X_{jr}) = (x_{j111},\ldots,x_{j1kl},\ldots,x_{jr11},\ldots,x_{jrkl})$, define a particular product variant P_j, where x_{jrkl} implies that the *l*-th instance of the *k*-th base module is selected for compound module m_r^C in product variant P_j. Note that x_{jrkl} is a binary variable, such that $x_{jrkl} = 1$ indicates that a compound module contains a module instance; and $x_{jrkl} = 0$ means not. Hence X_{jr} represents a choice decision vector of compound module m_r^C for product variant P_j. Then $X = (\vec{X}_1,\vec{X}_2,\ldots,\vec{X}_J)$ represents the decision vector of a product family, where the number of product variants J and the number of compound modules R are decision variables to be optimized.

The upper-level objective function should be consistent with the ultimate goal of PFA planning towards maximal customer satisfaction while leveraging enterprise profitability (Kaul and Rao, 1995). Customer satisfaction is usually evaluated according to the utility of product offerings. The utility of the *i*-th market segment for the *j*-th product variant is denoted as U_{ij}. It can be derived by a linear function of the part-worth utilities of the attribute levels of P_j, i.e $U_{ij} = \sum_{k \in K}\sum_{l \in L_k} w_{jk}u_{ikl}x_{jkl} + \pi_{ij} + \varepsilon_{ij}$, where u_{ikl} denotes the utility of the *l*-th instance of the *k*-th base module as perceived by the customers of market i; w_{jk} stands for the weight of the *k*-th base module in product variant P_j; π_{ij} relates to the composite utility of P_j by customers in market i; and ε_{ij} is an error term for each segment-product pair. There are a number of methods available to estimate regression utility weights and the constant through a given set of observed consumer choice data, for example, full-profile conjoint analysis, adaptive conjoint analysis, and experimental choice analysis (Lewis et al., 2006),

Evaluation of the manufacturer's profitability is essentially originated from fulfilment costs across multiple echelons of the supply chain. Jiao and Tseng (2004) propose a method for modeling the costs of providing variety based on variation of process capabilities. A process capability index thus lends itself to be an effective instrument for handling the sunk costs related to product families and the shared resources. A cost function, denotes as TC, can be formulated by aggregating respective cost elements of the supply chain, that is, $TC = (1 + \alpha\Psi(Y,Z))(TC^S + TC^M + TC^{Ap} + TC^D)$, where TC^s is the total cost associated with the suppliers; TC^M is the total cost of manufacturing plants involved in the supply chain; TC^{Ap} is the total cost of assembly plants; TC^D is the total cost of DCs, α is a lower-level profit rate used in the upper-level problem; and $\Psi(Y,Z)$ is the associated variable function determined by technology, quality, reliability, and other factors in the supply chain. Therefore, the mathematical model of PFA optimization can be formulated as a leader problem, as the following:

$$\max \quad F(x_{jrkl},J,R) = \frac{U}{TC} \tag{1.0}$$

$$s.t. \quad U = \sum_{i=1}^{I}\sum_{j=1}^{J}\left(\sum_{k \in K}\sum_{l \in L_k} w_{jk}u_{ikl}\left(\sum_{r=1}^{R} x_{jrkl}\right) + \pi_{ij}\right) \tag{1.1}$$

$$TC = (1 + \alpha\Psi(Y,Z))\left(TC^{S} + TC^{M} + TC^{Ap} + TC^{D}\right) \tag{1.2}$$

$$\sum_{r=1}^{R}\sum_{l=1}^{L_k} x_{jrkl} \leq 1 \tag{1.3}$$

$$\sum_{k=1}^{K}\sum_{l=1}^{L_k}\left|\sum_{r=1}^{R} x_{jrkl} - \sum_{r=1}^{R} x_{j'rkl}\right| > 0 \tag{1.4}$$

$$x_{jrkl} \in \{0,1\} \tag{1.5}$$

$$J,R \in N^{+} \tag{1.6}$$

$$\alpha \in (0,1) \tag{1.7}$$

$$\Psi(Y,Z) \in (0,1) \tag{1.8}$$

$$j = 1,\ldots,J, r = 1,\ldots,R, k = 1,\ldots,K, l = 1,\ldots,L_k \tag{1.9}$$

Eq. (1.0) describes the objective function for PFA planning by the utility per cost measure of all planned product variants in a family with respect to (x_{jrkl},J,R). Eq. (1.1) defines the total utility of a planned product family based on conjoint analysis. Eq. (1.2) determines the total fulfillment cost of all product variants by aggregating the costs of associated modules to be purchased, manufactured, assembled, and distributed through the supply chain. Eq. (1.3) constrains the XOR relationships between a base module and its instances. Eq. (1.4) enables difference in planned product variants. Eqs. (1.5-1.9) enforce all the decision variables and parameters within the range of the decision space.

For practical problems, specific technical compatibility constraints may be introduced as well. A common issue in PFA configuration is about compatibility constraints, which can be defined by restricting selection variable x_{jrkl}. For instance, the incompatibility between two base modules $m_{k_1}^{B}$ and $m_{k_2}^{B}$ can be expressed as: $\sum_{l=1}^{L_{k_1}} x_{jk_1l} + \sum_{l=1}^{L_{k_2}} x_{jk_2l} \leq 1$. Similarly, for the case where two base modules $m_{k_3}^{B}$ and $m_{k_4}^{B}$ must be included together in a product variant, an AND selection constraint can be introduced as: $\sum_{l=1}^{L_{k_4}} x_{jk_4l} \leq \sum_{l=1}^{L_{k_3}} x_{jk_3l}$.

5. OPTIMIZATION OF SUPPLY CHAIN CONFIGURATION

To fulfill each product variant in a PFA, the supply chain decision model deals with such optimal choice of suppliers for acquiring specific base modules, manufacturers for producing compound modules, assembly plants for assembling the product and DCs for delivering the product. The lower-level optimization model is formulated to address the respective supplier selection, manufacturer selection, assembly plant selection and DC selection problems.

5.1 Supplier Decision Model

The candidate suppliers for each specific base module (i.e., module instance, m_{kl}^{*}) are identified a priori through a company's supply contracting efforts. Corresponding to a module instance to be selected for a product variant configuration, the SCA decides a particular supplier for the module instance who runs an economic order quantity policy to maintain inventory supply for the module instance (Yang et al., 2015). Since the candidate suppliers differ in their operations, each supplier exhibits varying figures of raw material cost, inventory holding cost, transportation cost, and ordering cost. Therefore, the supplier selection problem is formulated according to the total inventory cost to be minimal, as the following,

$$\min \quad TC^{S} = \sum_{s=1}^{S_{kl}}\sum_{k=1}^{K}\sum_{l=1}^{L_k}\left(C_{kl}^{s}d_{m_{kl}^{*}} + RC_{kl}^{s}\frac{d_{m_{kl}^{*}}}{z_{kl}^{s}} + \frac{HC_{kl}^{s}z_{kl}^{s}}{2} + TCP_{kl}^{s}d_{m_{kl}^{*}}\right)y_{kl}^{s}f\left(y_{kl}^{s}\right) \tag{2.0}$$

$$s.t. \quad y_{kl}^{s} \le 1 - \prod_{j=1}^{J}\left(1 - \sum_{r=1}^{R}x_{jrkl}\right) \tag{2.1}$$

$$\sum_{s \in S_{kl}} y_{kl}^{s} = N_{kl}^{s} \tag{2.2}$$

$$d_{m_{kl}^{*}} = \sum_{i=1}^{I}\sum_{j=1}^{J}p_{ij}Q_i\left(\sum_{r=1}^{R}x_{jrkl}\right) \tag{2.3}$$

$$d_{m_{kl}^{*}} \le W_{kl}^{s} \tag{2.4}$$

$$0 \le z_{kl}^{s} \le d_{m_{kl}^{*}} \tag{2.5}$$

$$z_{kl}^{s} \le y_{kl}^{s}M \quad \text{M is a sufficiently large constant} \tag{2.6}$$

$$z_{kl}^{s} + 1 > y_{kl}^{s} \tag{2.7}$$

$$y_{kl}^{s} \in \{0,1\} \tag{2.8}$$

where for the s-th supplier, C_{kl}^{s} is the material cost per unit of base module instance m_{kl}^{*}; RC_{kl}^{s} is the ordering cost; HC_{kl}^{s} is the inventory holding cost per unit of m_{kl}^{*}; TCP_{kl}^{s} is the transportation cost per unit of m_{kl}^{*}; N_{kl}^{s} denotes the number of suppliers selected for supplying m_{kl}^{*}; W_{kl}^{s} is the supply capacity for m_{kl}^{*};

$f\left(y_{kl}^{s}\right)$ is the demand of m_{kl}^{*} proportionally allocated to the s-th supplier; and p_{ij} is the probability of demand of market segment i for product variant P_j;

Constraint Eq. (2.1) indicates whether or not a base module instance should be supplied. A specific base module is assumed to be supplied by a selected number of suppliers, as shown in constraint Eq. (2.2). Constraint Eq. (2.3) defines the total demand of m_{kl}^{*}. Constraint Eq. (2.4) restricts the demand for m_{kl}^{*} to be supplied within the capacities of the selected suppliers, whereas constraint Eq. (2.5) makes sure that a supplier's supply of m_{kl}^{*} is not more than the demand. Constraints Eq. (2.7-2.8) describe the compatibility between the choice variable of suppliers and the amount of supply with respect to the actual production - whenever the selection variable assumes a zero value, the supply amount must be zero; otherwise if the selection variable becomes one, the supply amount must be larger than zero.

5.2 Manufacturer Decision Model

The supply chain decisions for manufacturer selection account for the production of compound modules from base module instances. It is reasonable to assume that supply chain configuration at this echelon is enacted within established manufacturing plants. As for how to set up a new plant or develop vendors of manufacturing, it is not within the scope of supply chain configuration. Assume each compound module can be produced by more than one manufacturer. While all are capable to produce a compound module, these alternative manufacturers may perform differently in terms of operational costs, as well as material handling and logistics costs. It is common that the manufactures adopt a just-in-time strategy for their inventory management, in which the inventory-related costs are minimized (Yang et al., 2015). The manufacturers can also implement a vender managed inventory policy to include inventory costs through selling prices. The decision model for manufacturing plant selection can thus be formulated as:

$$\min\ TC^{M} = \sum_{r=1}^{R}\sum_{m=1}^{M_r}\left(C_r^m + \sum_{j=1}^{J}\left(\sum_{k=1}^{K}\sum_{l=1}^{L_k}\left(PC_{kl}^{m} d_{m_r^C} x_{jrkl} t_{kl}^{m}\left(z_r^m\right)\right) + TCP_r^m d_{m_r^C}\right)\right) y_r^m f\left(y_r^m\right) \tag{3.0}$$

$$s.t. \quad \sum_{m=1}^{M_r} y_r^m = N_r^m \tag{3.1}$$

$$d_{m_r^C} = \sum_{i=1}^{I} p_{ij} Q_i \tag{3.2}$$

$$d_{m_r^C} \le W_r^m \tag{3.3}$$

$$z_r^m \in \{0 \text{ or production methods set}\} \tag{3.4}$$

$$z_r^m \leq y_r^m M \quad \text{M is a sufficiently large constant} \tag{3.5}$$

$$z_r^m + 1 > y_r^m \tag{3.6}$$

$$y_r^m \in \{0,1\} \tag{3.7}$$

where for manufacturer m, C_r^m is the annual fixed cost allocated to compound module m_r^C; PC_{kl}^m is the unit variable cost for manufacturing a compound module m_r^C; TCP_r^m is the unit logistics cost of handling compound module m_r^C from manufacturer m to the assembly facility; $t_{kl}^m(z_{kl}^m)$ is the time of producing m_r^C by manufacturer m; $f(y_r^m)$ is the total demand proportionally allocated to manufacturer m; N_r^m is the maximal number of manufacturers selected for production of m_r^C; d_{jr} is the annual demand for m_r^C; and W_r^m is the production capacity of manufacturer m to produce m_r^C.

Constraint Eq. (3.1) indicates that the production of a compound module can be assigned to more than one manufacturing plant. While constraint Eq. (3.2) defines the demand for a compound module, constraints Eq. (3.3) makes sure that the capacity of each plant can satisfy the demand. Constraint Eq. (3.4) provides options of production capability. Constraint Eq. (5) guarantees that the selection variable for manufacturer m takes a zero value when it is not assigned to produce m_r^C. Constraint Eq. (3.6) enforces that a selected manufacturer m must produce at least one type of compound module. Constraint Eq. (3.7) ensures that every compound module will be produced.

5.3 Assembly Decision Model

The advantage of product family configuration lies in assembly-to-order production that enables mass customization. The assembly plants could be independent from manufacturing by outsourcing module manufacturing to vendors. Among multiple alternatives, one assembly plant needs to be selected for assembling of compound modules from manufacturing plants into end-product variants. Different candidate plants for assembling may possess different capacities in terms of processing time, labor cost, and the like. The total cost at an assembly plant comprises the operational fixed and variable costs, along with the transportation cost. Therefore, the decision model for assembly deals with these cost components subject to engineering constraints, that is,

$$\min \mathrm{TC}^{Ap} = \sum_{Ap=1}^{AP_j} \sum_{j=1}^{J} \left(C_j^{Ap} + \sum_{r=1}^{R} \left(PC_{jr}^{Ap} d_{P_j} t_r^{Ap}\left(z_j^{Ap}\right)\right) TCP_j^{Ap} d_{P_j} \right) y_j^{Ap} f\left(y_j^{Ap}\right) \tag{4.0}$$

s.t.

$$\sum_{Ap \in AP_j} y_j^{Ap} = N_j^{Ap} \tag{4.1}$$

$$d_{P_j} = \sum_{i=1}^{I} p_{ij} Q_i \tag{4.2}$$

$$d_{P_j} \leq W_j^{Ap} \tag{4.3}$$

$$z_j^{Ap} \in \{0 \text{ or production methods set}\} \tag{4.4}$$

$$z_j^{Ap} \leq y_j^{Ap} M \quad \text{M is a sufficiently large constant} \tag{4.5}$$

$$z_j^{Ap} + 1 > y_j^{Ap} \tag{4.6}$$

$$y_j^{Ap} \in \{0,1\} \tag{4.7}$$

The above assembly decision model is defined for each assembly plant Ap, for which C_j^{Ap} is the annual fixed operational cost allocated to product variant P_j; PC_{jr}^{Ap} is the unit variable cost for product variant P_j; TCP_j^{Ap} is the unit transportation cost of product variant P_j to be moved from Ap to the DC; $t_r^{Ap}\left(z_j^{Ap}\right)$ is the processing time for assembling a unit of product variant P_j; $f\left(y_j^{Ap}\right)$ denotes the demand proportionally allocated to Ap; N_j^{Ap} indicates the number of assembly plants selected in the supply chain; d_j is the annual demand for product variant P_j; and W_j^{Ap} is the production capacity of assembly plant Ap for producing P_j.

Constraint Eq. (4.1) restricts the number of selected assembly plants, whilst constraint Eq. (4.2) defines the total demand of product variant P_j. Constraint Eq. (4.3) denotes the capacity limit of each assembly plant and Eq. (4.4) indicates the selection of a production method (i.e., assembly planning). Constraint Eq. (4.5) guarantees that when an assembly plant is not selected, the corresponding production method should be null. Eq. (4.36) is constraint that makes sure when an assembly plant is selected, the corresponding production must be greater than zero.

5.4 Distribution Center Decision Model

The final echelon of the product family supply chain is associated with DCs for the end-product variants to be delivered to retailers. The decision model for DCs aims to minimize total transportation and inventory costs occurring at DCs, which include the setup cost of DCs, the inventory costs at each DC and the transportation cost from on DC to the retailers. Assume that all the DCs have the capabilities to deliver any of product variants in a product family. Then the decision model for DCs can be formulated as the following:

$$\min \quad TC^D = \sum_{d=1}^{D}\left(\sum_{j=1}^{J}\left(RC_j^d\frac{d_{P_j}}{z_j^d} + TCP_j^d d_{P_j} + \frac{HC_j^d z_j^d}{2}\right)\right)y^d f\left(y^d\right) \tag{5.0}$$

$$s.t. \quad \sum_{d \in D} y^d = N^D \tag{5.1}$$

$$d_{P_j} = \sum_{i=1}^{I} p_{ij} Q_i \tag{5.2}$$

$$d_{P_j} \le W_j^d \tag{5.3}$$

$$0 < z_j^d \le d_j \tag{5.4}$$

$$y^d \in \{0,1\} \tag{5.5}$$

where the decision variables are the selection variable y^d from a set of DC alternatives and the optimal order quantity z_j^d. In addition, RC_j^d is the ordering cost of product variant P_j at the d-th DC; TCP_j^d denotes the unit transportation cost of product variant P_j to be delivered from d to retailers; and HC_j^d represents the unit inventory holding cost for P_j at the d-th DC. Eq. (5.1) restricts the number of DCs selected for the product family to be capped at N^d. Constraint Eq. (5.2) represents the total demand of product variant P_j. Constraint Eq. (5.3) expresses the capacity limit at DCs. Eqs. (5.4) and (5.5) enforce the parameters ranges of two decision variables.

6. Stackelberg Game Model for Leader-Follower Joint Optimization

Both PFA planning and SCA decisions involve many different problem domains and are associated with multiple decision makers that have to compromise for conflicting goals in order to maximize each individual's own payoffs. While the goal of optimal PFA planning is mainly geared toward maximal customer-perceived utility per cost (Jiao et al., 2007), SCA optimization aims at minimal costs at each echelon of the supply chain. As such, coupling of PFA and SCA issues leads to a non-cooperative game that needs to arrive at equilibrium solutions between two different configuration optimization problems. For joint optimization of PFA and SCA configuration, the PFA planning problem plays a dominant role and the SCA problem acts as the feedback, exemplifying a paradigm of leader-follower game theoretic decision making. Fig. 3 illustrates the decision variable structure of the one leader-four follower optimization model, in which the upper-level problem controls the lower-level problems through PFA decision variables X and the lower-level problems feedback the upper level through the respective supple chain decision variables (Y^S,Z^S), (Y^M,Z^M), (Y^{AP},Z^{AP}), and (Y^D,Z^D).

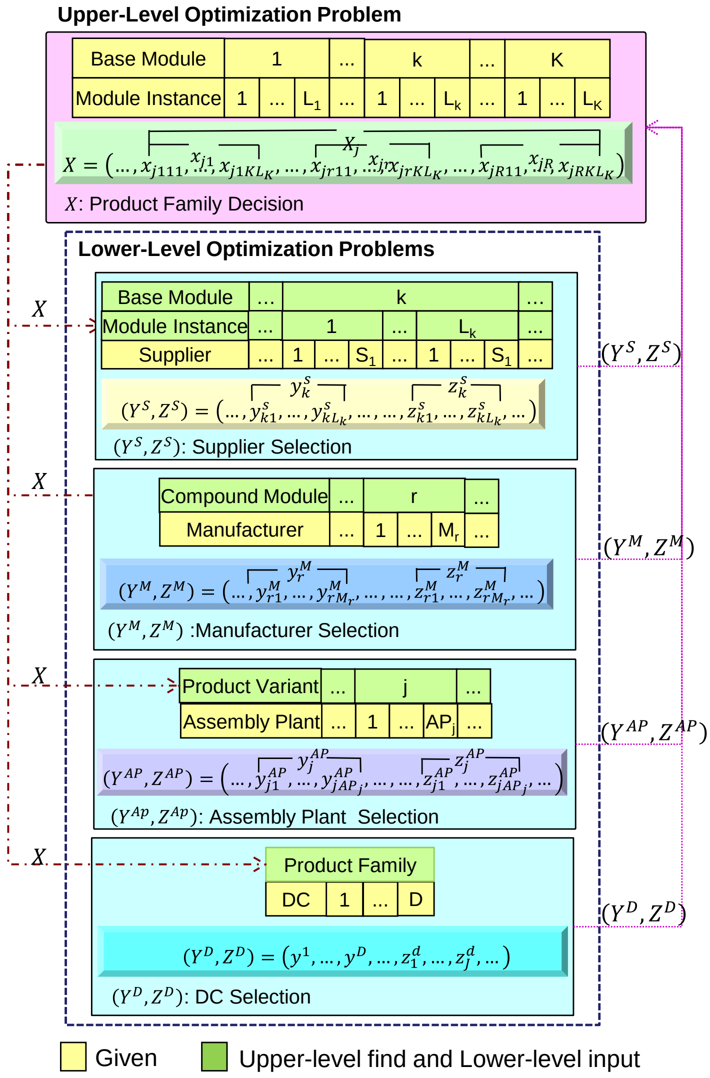

Fig. 3: Model decision variable structure

In line with Stackelberg game theoretic decision making, joint optimization of product family architecting and supply chain configuration can be formulated as a leader-follower bilevel optimization model. The PFA planning optimization problem acts as a leader, constituting an upper-level optimization model. The supply chain configuration problem comprises the supplier, manufacturer, assembly, and DC decision models. Each of these decision models performs as a follower and entails a specific lower-level optimization model. Both the leader for PFA planning and the follower for SCA configuration possess their own strategies (in terms of decision variables) and payoffs (in terms of objective functions). The joint optimization process is initiated by the leader for PFA planning who enforces its strategies on the lower-level problems. The followers responds to the upper-level PFA planning decisions by choosing its own strategies to optimize its own objective function regarding the respective supplier, manufacturing, assembly and DC decisions. Such a 1-leader-4-follower bilevel joint optimization model can be summarized as follows:

$$\max \quad F(x_{jrkl}, J, R) = \frac{U}{TC} \tag{6.0}$$

$$s.t. \quad \text{Constraint Eqs.(1.1) - (1.9)} \tag{6.1}$$

$$\min \quad TC^{s}\left(y_{kl}^{s}, z_{kl}^{s}\right) \tag{6.2}$$

$$s.t. \quad \text{Constraint Eqs.(2.1) - (2.8)} \tag{6.3}$$

$$\min \quad TC^{M}\left(y_{r}^{m}, z_{r}^{m}\right) \tag{6.4}$$

$$s.t. \quad \text{Constraint Eqs.(3.1) - (3.7)} \tag{6.5}$$

$$\min \quad TC^{AP}\left(y_{j}^{Ap}, z_{j}^{Ap}\right) \tag{6.6}$$

$$s.t. \quad \text{Constraint Eqs.(4.1) - (4.7)} \tag{6.7}$$

$$\min \quad TC^{D}\left(y^{d}, z_{j}^{d}\right) \tag{6.8}$$

$$s.t. \quad \text{Constraint Eqs.(5.1) - (5.5)} \tag{6.9}$$

Fig. 4 illustrates the interactive decision making process of the bilevel joint optimization model. First, PFA planning makes optimal decisions about the number of product variants, the number of compound modules and selection of base module instances, embodied by decision variables J, R, and x_{jrkl}, respectively. These results of PFA planning decisions then become the parameters for lower-level optimization of supply chain configuration through minimizing total supply chain-related costs. By such a parametric optimization process, the supply chain configuration problem responds to the upper level by providing feedback on the fulfillment costs of a product family including sourcing costs of base module instances, manufacturing costs of compound modules, assembly costs of product variants, and delivery costs of end-products. The impact of supply chain decisions on product family architecting is achieved through the upper-level optimization problem to adjust its PFA planning decisions according to the lower-level supply chain configuration feedback on the fulfillment costs of product variants while maximizing the customer-perceived utility of the overall product family. The bilevel joint optimization process proceeds in an iterative manner until Stackelberg equilibrium solutions are reached for both the leaders and the followers.

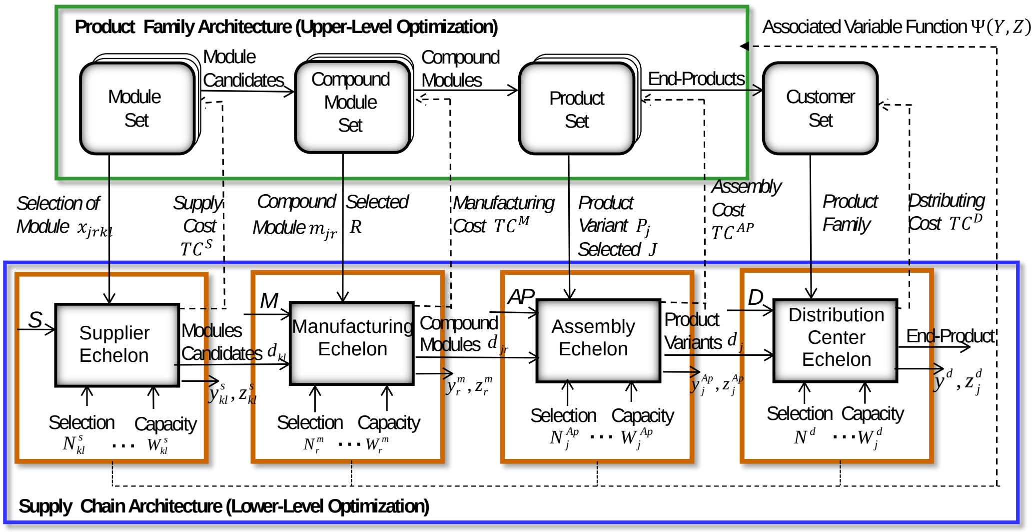

Fig. 4: Interaction decision making between the upper- and lower-level optimization problems

7. Model Solution

7.1 Nested Genetic Algorithms

The bilevel joint optimization model entails a nonlinear, mixed integer program that is NP-hard. Existing bilevel optimization solutions are generally categorized as direct and indirect methods. The indirect approach aims to convert the original bilevel problem to a single-level program, such as KKT conditions and penalty function methods. The direct approach on the other hand is aligned with the bilevel decision mechanism, such as satisfactory solution methods. However, it becomes challenging for direct solution methods to tackle large size optimization problems. For instance, the larger the number of supply chain nodes for fulfilling modules and its instances, the less efficient of the direct solution methods.

Product family and supply chain configuration problems essentially entail combinatorial optimization, for which genetic algorithms are proven to be advantageous for large and complex problems (Oliveto et al., 2007). In our model Eqs. (6.0)-(6.9), each of the upper-level and lower-level optimization problems is associated with different engineering characteristics, which makes the assumptions of KKT conditions hardly hold. We thus propose to employ a nested scheme of multiple GAs to address the interactive decision making between the upper- and lower-level problems, whilst solving each individual leader's or follower's problem with one specific GA. The nested GA process reveals the underlying coupling between the leader and the followers through the dominant and feedback variables that are solved by satisfying all constraints of the bilevel model and limiting the optimal solutions within the feasible region of the design space.

As shown in Fig. 5, the nested GA solution process starts with the upper-level GA by generating an initial population (minimal values of *J* and *R*, along with module selection X) that is feasible for the leader's problem. To propagate this initial solution for the dominant variables to the followers' problems, each decision vector X from the upper level becomes parameters of each individual lower-level GA that is to be solved locally within the design space of each lower-level optimization problem for supplier selection ($Y^S(X)$ and $Z^S(X)$), manufacturer selection ($Y^M(X)$ and $Z^M(X)$), assembly selection ($Y^{Ap}(X)$ and $Z^{Ap}(X)$), or DC selection ($Y^D(X)$ and $Z^D(X)$). All the lower-level optimal solutions, (Y^{S*}, Y^{M*}, Y^{Ap*}, Y^{D*}) and (Z^{S*}, Z^{M*}, Z^{Ap*}, Z^{D*}), then perform as the feedback variables to re-run the upper-level GA and in turn to adjust the upper-level solution (J^* and R^*). This process iterates until both and leader and followers reach a Stackelberg game equilibrium.

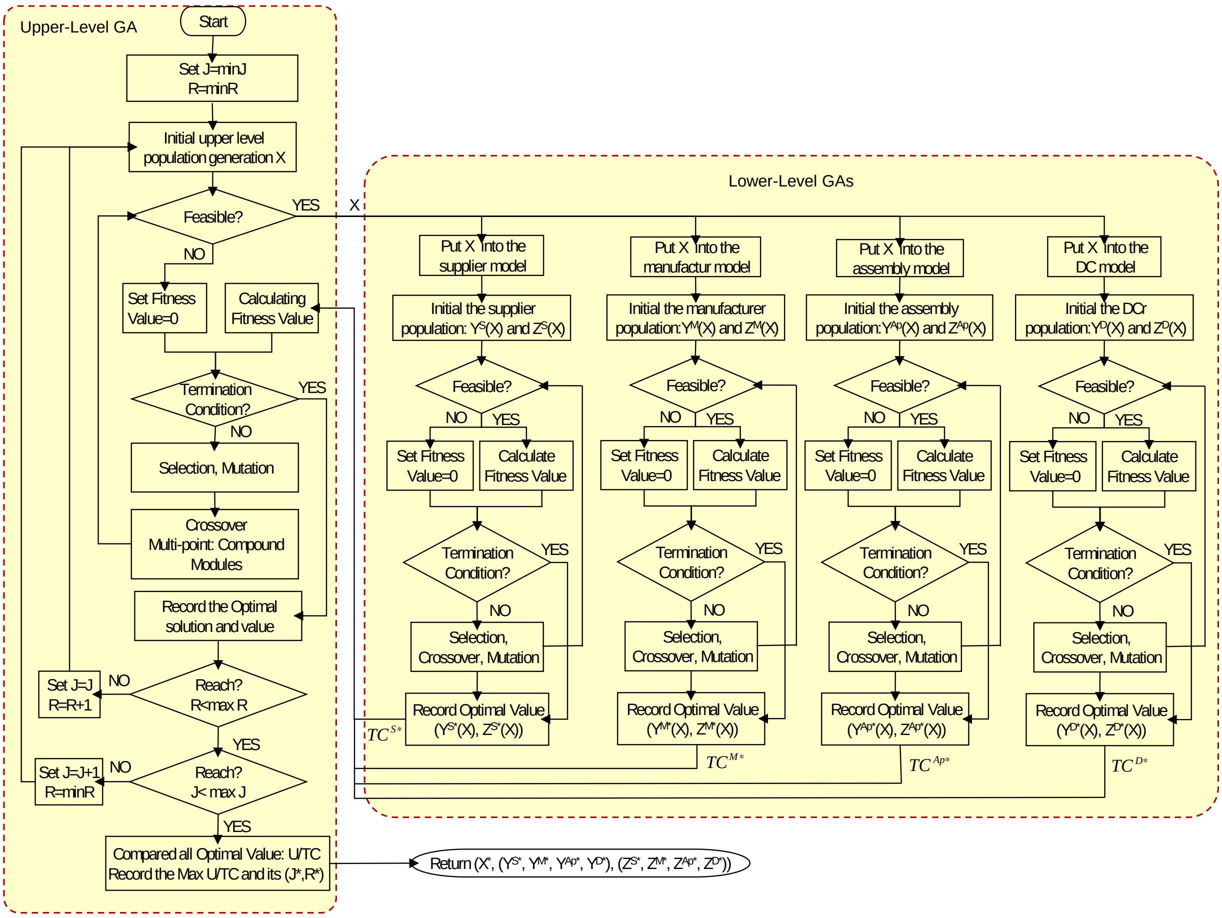

Fig. 5: Nested GA solution for leader-follower bilevel joint optimization

7.2 GA Encoding and Operators

The critical issue of GAs is the encoding of the semantics of each decision variable of bilevel optimization by a string of digits, namely a chromosome. As illustrated in Fig. 6, we propose to encode a product family using a generic chromosome that comprises fragments (i.e., sub-string), such that each fragment represents a product variant of the product family. Likewise, each fragment can be further fragmented such that each sub-fragment corresponds to a compound module. Every element of the string, namely a gene, then denotes a particular base module.

Selection of base module instances can be encoded in a similar fashion. As show in Fig. 7, all the based modules are listed as fragments of the chromosome, whilst each fragment represents all the instances of a base module and the length of the fragment corresponds to the total number of instance of that particular base module. A binary decision vector, $v = \left[v_{11}, \ldots, v_{kl}, \ldots v_{KL_K}\right]$, can be introduced to represent the XOR selection of one instance for every based module. Each gene denotes the selection ($v_{kl} = 1$) or not ($v_{kl} = 0$) of a base module particular instance (m^*_{kl}).

Supplier selection for sourcing of a particular base module instance can be encoded similarly using a fragmented chromosome and a selection decision vector$[Y^S, Z^S]$, as shown in Fig. 8. Using the same method of generic encoding, three separate GAs can be constructed to represent the manufacturer, assemble and DC decision models, respectively.

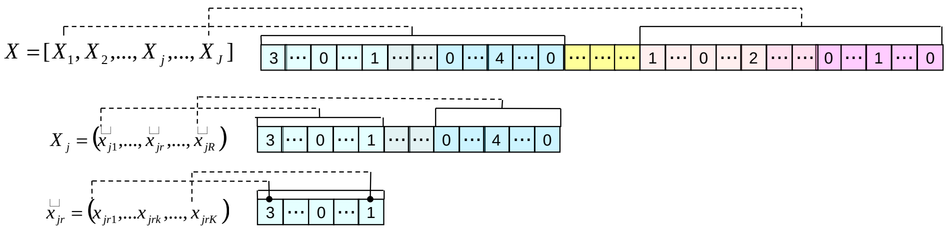

Fig. 6: Product family chromosome encoding

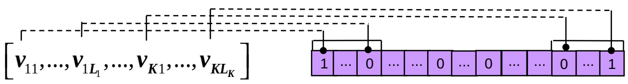

Fig. 7: GA encoding of base module instance selection

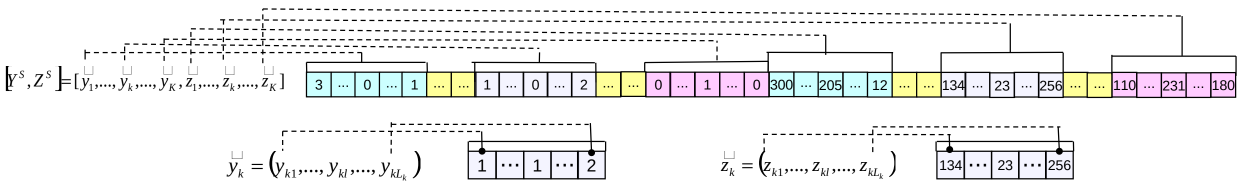

Fig. 8: Supplier selection chromosome encoding

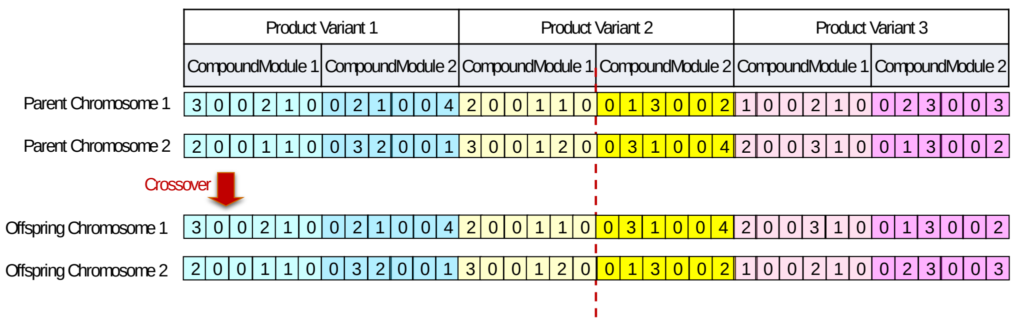

Fig. 9: GA encoding scheme for crossover node selection

Considering convergence performance the configuration compatibility constraints of the bilevel model, the nested GAs adopt a multi-point crossover operator, so as to always encapsulate the genes of a compound module always as one unit. Fig. 9 illustrates how the multi-point crossover method manages to control the nodes of crossover operation to coincide with the boundaries of chromosome fragments for compound modules.

A standard mutation process of generating offspring after crossover is adopted for the nested GAs. The mutation process randomly picks a gene within each string using a small probability (referred to as mutation rate) and alters the corresponding attribute levels at random. This process enables a small amount of random search, and thus ensures that the nested GA search does not quickly converge at a local optimum. The processes of crossover and reproduction are repeated until all the upper- and lower-level populations converge or all the GAs reach a pre-specified number of generations. The threshold for each GA can be specified a priori based on the specific contexts of the problem domain, or determined through experiment studies of computational performance of the algorithms.

8. CASE STUDY

8.1 Power Transformer Product Family and Supply Chain

A case study of power transformer product family architecting is conducted in a company that involves assembly-to-order production through a multi-site manufacturing supply chain, as shown in Fig. 10. The focus is oil-immersed power transformers that typically consist of a tank body (core, windings, insulation, and power leads), transformer oil, conservator, cooling tubes, breather, explosion vent, bushing, etc. Table 1 summarizes the base modules that are to be acquired from the suppliers. For the product family under study, there are multiple instances for each base module and specific modules instances are coded in Table 1.

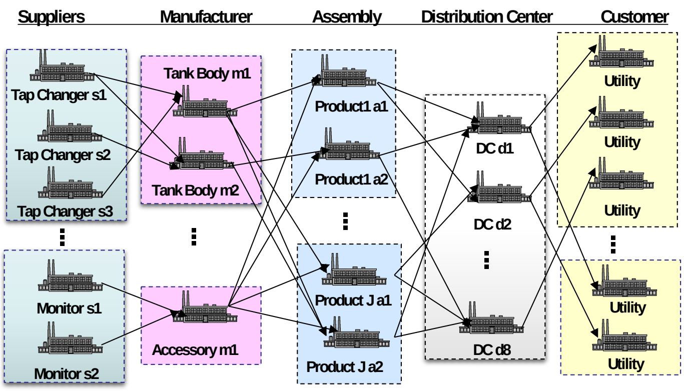

Fig. 10: Multi-site manufacturing supply chain for power transformer product family

Table 1: Power transformer base modules and their instances

ID	Base	Instanc	ID	Base Module	Instanc	ID	Base	Instanc

	Module	e
m1	Iron core	m11
		m12
		m13
		m14
m2	Winding	m21
m3	Tap changer	m31
		m32
		m33
		m34

		e
m4	Tank shell	m41
		m42
		m43
m5	Transformer oil	m51
m6	Cooling unit	m61
		m62
		m63
m7	High/low-voltage bushing	m71
		m72
		m73
		m74

	Module	e
m8	Oil conservator	m81
		m82
m9	Monitor	m91
		m92
		m93
		m94
m10	Standard Accessories	m101

8.2 PFA Planning

For illustrative simplicity without losing generality, we consider one leading market for the product family, i.e. $I = 1$. Also assume that $Q_1 = 10000$ and p_{ij} are of equal choice probabilities. Based on the company's market research, the initial number of product variants can be set as $J = \{2,3\}$, and the number of compound modules is set as $R = \{2,3,4\}$. To specify utility function U_{ij} in the upper-level optimization model, w_{jk} is obtained based on the company's market survey; u_{ikl} is determined through conjoint analysis; and π_{ij} is derived from the composite utility of P_j by S_i. The upper level allocates a profit to the lower level by a profit rate, $\alpha = 0.1$, whereas the lower level sets associated variable function as $\Psi(Y,Z) = 0.9$.

Customization of the transformers are defined through the specification of functional attributes and options for the attribute values. Table 2 summarizes the mapping relationships between various attribute options to the corresponding base module instances. Customer-perceived utilities of different attribute options of the transformers are determined by compiling partworth utility of each based module instance that is established through conjoint analysis. The partworth utilities of based module instances are summarized in Table 3. Given various options of 8 voltage ratings * 3 tank materials * 3 cooling unit * 4 monitor * 4 top changer * 2 oil conservator, transformer customization encompasses a total number of *J* = 9216 possible combinations. Using the Taguchi Orthogonal Array Selector provided in SPSS software, a total number of 20 orthogonal product profiles are generated for experiment setup of conjoint analysis, as shown in Table 4. With these profiles, a fractional factorial experiment is designed to explore customer preferences, for which 40 customers are invited as the respondents.

Table 2: Mappings of transformer attributes to based module instances

Attributes	Attribute Value	Base Module Instance

Table 3: Partworth utilities of module instances

Base Module	Module Instance	Partworth Utility

Rated voltage / Rated capacity	Low-voltage	m11
	Middle-voltage	m12
	High-voltage	m13
	Special-voltage	m14
	Low-voltage	m31 UBB
	Low-voltage	m32 UCG
	High-voltage	m33 LTC
	Special-voltage	m34 UCL
…	…	…
Oil circulation way	Non	m81
	Have	m82
		m51

		u_{ikl}
m1	m11	1.85
	m12	1.48
	m13	1.59
	m14	1.63
m2	m21	3.89
…	…	…
m9	m91	1.46
	m92	-0.85
	m93	-0.66
	m94	2.3
m10	m101	4.6

Optimal PFA planning requires domain experience to specify the maximal number of compound modules, which essentially involves a granularity issue of modular design (Jiao et al., 2007). Once the granularity of compound modules are specified a priori, the grouping of based modules can be accordingly determined. Tables 5, 6 and 7 show the corresponding results of grouping based modules into different compound modules depending on the pre-specified numbers of compound modules at R=2, 3, and 4, respectively. In practice, identification of compound modules is much domain dependent, subject to engineering requirements, functionality, as well as module compatibility constraints.

Table 4: Orthogonal setup of product profiles for conjoint analysis

ID	Iron core	Winding	Tap changer	Tank shell	Transformer oil	Cooling unit	High/low-voltage bushing	Oil conservator	Monitor	Standard accessories
1	small	same	UBB	Steel material	same	self-cooling	EW02	Non	the status of Tap Changer	same
2	small	same	UBB	Steel material	same	air cooling	RK-S06	have	the status of Tap Changer	same
...	...	...	...	...	...	...	...	...	...	...
32	unset	same	LTC	Steel material	same	air cooling	RK-S06	have	the status of cooling functions	same

Table 5: Identification of compound modules (*R=2*)

Compound model	**Base model**
Tank body	Iron core, winding, Tank shell is must to choice, the other is obtained by optimization
Other attachments	Cooling unit, oil conservator is must to choice, the other is obtained by optimization

Table 6: Identification of compound modules (*R=3*)

Compound model	**Base model**
Tank body	Iron core, winding is must to choice, the other is obtained by optimization
Oil tank	Tank shell, Transformer oil is must to choice, the other is obtained by optimization
Other attachments	Cooling unit, oil conservator is must to choice, the other is obtained by optimization

Table 7: Identification of compound modules (*R=4*)

Compound model	Base model
Tank body	Iron core, winding is must to choice, the other is obtained by optimization
Oil tank	Tank shell, Transformer oil is must to choice, the other is obtained by optimization
Outlet device	High/low-voltage bushing is must to choice, the other is obtained by optimization
Other attachments	Cooling unit, oil conservator is must to choice, the other is obtained by optimization

8.3 Supply Chain Configuration

The supply chain data of the transformer product family are listed in Tables 8, 9, 10, and 11 for the related suppliers, manufacturers, assembly plants, and DCs, respectively. It is practical to select more than one supply chain entity for fulfilling the corresponding order (i.e., to reserve backup). Based on the company's supply chain performance study, Table 12 summarizes the numbers of backups planed for selection of the suppliers, manufacturers, assembly plants and DCs, respectively. In addition, order fulfillment at manufacturing and assembly is associated with options of process alternatives as well, as summarized in Tables 13 and 14.

Table 8: Available suppliers for module instances

ID	SC echelon	Module instance	Options	Capacity	Material cost	Ordering cost	Inventory cost	Transportation cost
S11	Supplier	m_{11}	1	40000	250	85	0.16	7
			2	20000	234	97	0.15	5
S12	Supplier	m_{12}	1	34000	270	90	0.2	8
			2	26000	289	99	0.16	6
			3	37000	276	110	0.17	8
…	…	…	…	…	…	…	…	…
S94	Supplier	m_{94}	1	56000	230	140	0.17	6
			2	34000	225	150	0.18	7
S101	Supplier	m_{101}	1	35000	395	480	0.21	6
			2	49000	381	500	0.23	6
			3	20000	379	530	0.2	10

Table 9: Available manufacturers for producing compound modules

ID	SC echelon	Compound module	Options	Capacity	Operational cost	Processing cost	Transportation cost
M1	Manufacturer	M1	1	72000	6700	430	9
			2	36000	7500	440	7
			3	43000	6900	410	8
…	…	…	…	…	…	…	…
M4	Manufacturer	M4	1	32000	8200	430	7
			2	53000	7600	460	5
			3	56000	7900	480	9

Table 10: Available assembly plants for producing product variants

ID	SC echelon	Product	Optio	Capacity	Fixed	Processing	Transportation

		variant	ns		cost	cost	cost
AP1	Assembly Plant	P1	1	5300	5000	440	7
			2	5700	6500	430	5
			3	5900	5400	470	9
AP2	Assembly Plant	P2	1	5500	6700	470	6
			2	5700	7000	430	7
			3	6100	6600	500	5
AP3	Assembly Plant	P3	1	5800	8300	490	8
			2	5300	7700	470	6

Table 11: Available DCs for delivering the end-products

ID	SC echelon	Options	Capacity	Ordering cost	Inventory cost	Transportation cost
D	DCs	1	12000	95	0.3	20
		2	17000	90	0.32	24
		3	14000	112	0.3	18
		4	18000	86	0.31	22
		5	16000	103	0.35	19
		6	21000	100	0.32	24
		7	19500	88	0.35	27
		8	20000	105	0.33	24

Table 12: Backup plan for selection of supply chain entities

Name/ID	Number	Name/ID	Number	Name/ID	Number	Name/ID	Number
Supplier S11	1	Supplier S41	1	Supplier S73	1	Manufacturer M1	2
Supplier S12	1	Supplier S42	1	Supplier S74	1	Manufacturer M2	2
Supplier S13	1	Supplier S43	1	Supplier S81	2	Manufacturer M3	1
Supplier S14	1	Supplier S51	3	Supplier S91	1	Manufacturer M4	1
Supplier S21	2	Supplier S61	1	Supplier S92	1	Assembly Plant AP1	2
Supplier S31	1	Supplier S62	1	Supplier S93	1	Assembly Plant AP2	1
Supplier S32	1	Supplier S63	1	Supplier S94	1	Assembly Plant AP3	1
Supplier S33	1	Supplier S71	1	Supplier S101	3	DC	3
Supplier S34	1	Supplier S72	1				

Table 13: Alternative process plans at each manufacturer

ID	SC echelon	Compound module	Options	Process plan	Processing time
M1	Manufacturer	M1	1	w1	4.3
				w2	5.1
			2	w1	4.4
				w2	4,8
			3	w1	4.1
				w2	5.2
…	…	…	…	…	…
M4	Manufacturer	M4	1	w1	4.3
				w2	5.15
			2	w1	4.6

			w2	4.92
		3	w1	4.8
			w2	5.35

Table 14: Alternative assembly plans at each assembly plant

ID	SC echelon	Options	Assembly plan	Processing time
AP1	Assembly Plant	1	w1	4.4
			w2	3.7
		2	w1	4.3
			w2	3.6
		3	w1	4.7
			w2	3.2
AP2	Assembly Plant	1	w1	4.7
			w2	6.1
		2	w1	4.3
			w2	3.15
		3	w1	5.0
			w2	4.75
AP3	Assembly Plant	1	w1	4.9
			w2	6.25
		2	w1	3.1
			w2	4.7

8.4 Results of Stackelberg Game Joint Optimization

The near-optimal solutions for joint transformer PFA planning and supply chain configuration are determined by running the nested GAs. For computational efficiency and meaningful problem contexts, the population size is capped at as 100 and the GAs adopt a crossover probability of 0.8 and a mutation probability of 0.01. The optimal values are obtained by enumerating all possible (J,R) as shown in Table 15.

Fig. 11 shows the convergence of the upper-level GA with respect to different settings of(J,R). The best upper-level fitness is achieved corresponding the scenario of J = 3 and R = 3. Fig. 12 shows the tradeoffs between the upper-level GA fitness for utility-to-cost ratio and the lower-level GAs for the supplier, manufacturing, assembly and DC costs. Upon convergence at around 130-th generation, the nested GAs return the optimal results of transformer product family architecting and the corresponding supply chain configurations. Table 16 shows the results of joint optimization. The corresponding results of transformer PFA and the supply chain configuration are interrupted in Tables 17, 18, and 19.

Table 15: Optimal values corresponding to different settings of (J,R)

(J ,R)	J=2□R=2	J=2□R=3	J=2□R=4	J=3□R=2	J=3□R=3	J=3□R=4
Utility/Cost(10^{-4})	26.4773	27.8323	29.1658	33.3930	34.1027	33.2482
TCS (10^{7})	3.5447	3.3337	3.1470	2.9101	2.7992	2.9705
TCM (10^{7})	2.1123	2.9603	3.0573	2.5283	2.2134	2.3442

TCAP (10^7)	2.3411	2.4002	2.3947	2.3140	2.3247	2.4346
TCD (10^7)	1.6363	1.9853	1.8385	1.9847	1.8296	1.9343

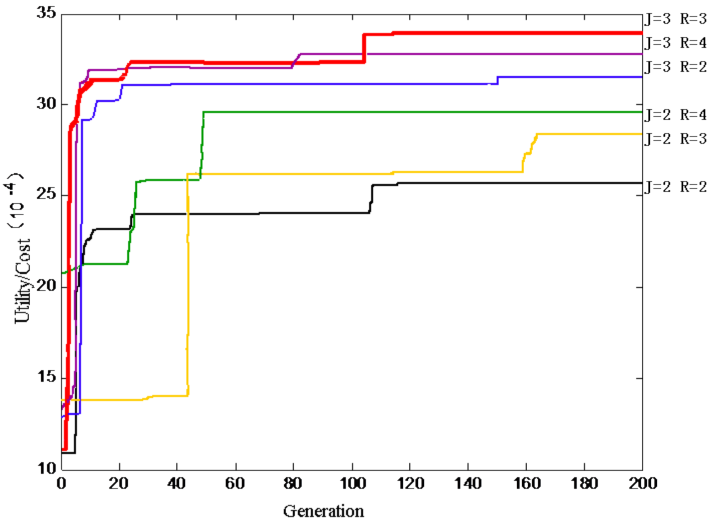

Fig. 11: Upper-level GA fitness with respect to different settings of (J, R)

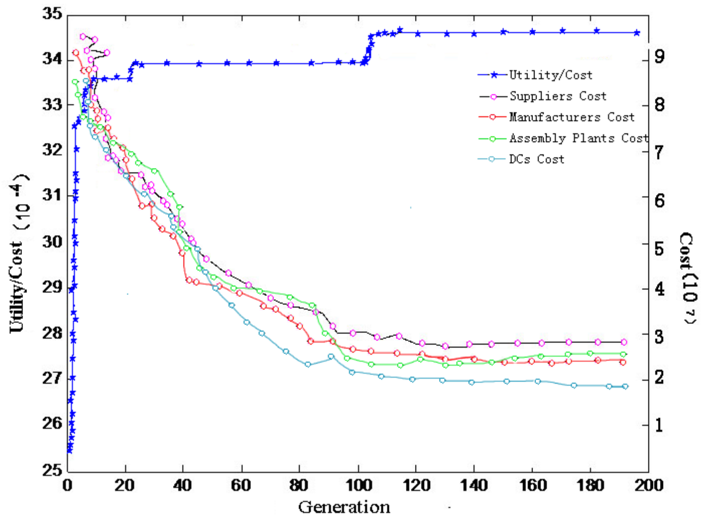

Fig. 12: Convergence of nested GAs corresponding to (J=3, R=3)

Table 16: Optimal solutions of transformer product family and supply chain configuration (J=3, R=3)

PFA	X=[3 1 2 0 0 0 0 0 0 0 0 0 0 0 0 2 1 0 0 0 0 0 0 0 0 0 0 0 0 3 3 1 2 1, 1 1 3 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 2 4 2 3 1, 2 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 3 1 0 0 0 0 0 0 0 0 0 0 0 1 1 2 1 1]
Utility/Cost	34.1027×10^{-4}
Suppliers	[Y, Z] =[1 1 2 0 1 3 2 3 1 0 2 1 1 1 2 3 0 1 3 1 0 2 3 1 2 2 0 0 3 1 2 3 271 351 321 0 712 707 234 315 267 0 259 278 265 712 659 693 0 365 277 357 0 315 273 345 521 472 0 0 481 765 691 735]
Supplier Cost	2.7992×10^{7}
Manufacturers	[Y, Z]=[1 1 0 0 1 1 0 1 0 , 0 1 0 1 1 0 0 2 1 0 0 0]
Mfg. Cost	2.2134×10^{7}
Assembly Plats	[Y, Z]=[1 0 1 0 1 0 0 1 , 1 0 0 1 1 0 1 0]
Assembly Cost	2.3247×10^{7}
DCs	[Y, Z]=[0 1 0 0 1 0 1 0 , 631 589 674]
DC Cost	1.8296×10^{7}

Table 17: Optimal plan for transformer PFA

Product Variant	P1	P2	P3
CompoundModule1 (Tank Body)	Iron core m13 Winding Tap changer UCG	Iron core m11 Winding Tap changer LTC	Iron core m12 Winding Tap changer UBB
CompoundModule2 (Oil tank)	Tank shell Steel material Transformer oil	Tank shell Steel material Transformer oil	Tank shell Steel material Transformer oil
CompoundModule3 (Other attachments)	Air cooling High/low-voltage bushing RK-S06 Without oil conservator A monitoring component for detecting the temperatures of the body Standard Accessories	Water cooling High/low-voltage bushing DW-N6 With oil conservator A monitoring component for detecting the status of cooling functions Standard Accessories	Self-cooling High/low-voltage bushing EGL02 With oil conservator A monitoring component for detecting the status of Tap Changer Standard Accessories

Table 18: Optimal configuration for the suppliers and DCs

Name/ID	Number	Optimal choice	Order quantity	Name/ID	Number	Optimal choice	Order quantity
Supplier S11	1	1	271	Supplier S62	1	1	365
Supplier S12	1	1	351	Supplier S63	1	3	277
Supplier S13	1	2	321	Supplier S71	1	1	357
Supplier S14	1	0	0	Supplier S72	1	0	0
Supplier S21	2	1 3	712 707	Supplier S73	1	2	315
Supplier S31	1	2	234	Supplier S74	1	3	273
Supplier S32	1	3	315	Supplier S81	2	1 2	345 521
Supplier S33	1	1	267	Supplier S91	1	2	472
Supplier S34	1	0	0	Supplier S92	1	0	0
Supplier S41	1	2	259	Supplier S93	1	0	0
Supplier S42	1	1	278	Supplier S94	1	3	481
Supplier S43	1	1	265	Supplier S101	3	1 2 3	765 691 735
Supplier S51	3	1 2 3	712 659 693	Distribute Center D	3	2 5 7	631 589 674
Supplier S61	1	0	0				
Supplier S51	3	1 2 3	712 659 693	Distribute Center D	3	2 5 7	631 589 674
Supplier S61	1	0	0				

Table 19: Optimal configuration for the manufacturing and assembly plants

Name/ID	Number	Optimal choice	Process plan	Processing time
Manufacturer M1	2	1 2	2 2	5.1 4.8
Manufacturer M2	2	2 3	1 2	4.3 5.3
Manufacturer M3	1	2	1	4.5
Manufacturer M4	1	0	0	0
Name/ID	**Number**	**Optimal choice**	**Assembly plan**	**Processing time**
Assembly plant AP1	2	1 3	1 2	4.4 3.2
Assembly plant AP2	1	2	1	4.3
Assembly plant AP3	1	2	1	3.1

8.5 Performance Analysis

Performance of the proposed leader-follower Stackelberg (LFS) game decision model is shown in Table 20, in terms of the achieved utility-to-cost ratio of the PFA and the associated supplier, manufacturing, assembly and DC costs. To demonstrate the advantages of the LFS model, computational experiments are set up to compare its performance with two traditional optimization methods: (1) non-joint optimization (NJOP) - independent optimization without considering the supply chain, and (2) an integrated method by "all-in-one" (AIO) multi-objective optimization.

The NJOP model first optimizes product family architecting alone to the maximal benefit, and then at the second stage minimizes the supply chain cost at each echelon independent of the results of the first stage. It basically treats PFA planning and its supply chain configuration as two separate optimization problems and solves them in isolation. The performance of the NJOP model is shown in Table 21. The AIO model optimizes PFA planning and supply chain configuration simultaneously by aggregating these two optimization problems into one single-level objective function. Table 22 shows the performance of the AIO model. Comparison of these results suggests that both the LFS and NJOP models achieve the optimal solutions given the same number of product variants and compound modules contained in the product family.

In terms of PFA performance by the utility-to-cost ratio measure, Fig. 13 shows the LFS model outperforms the NJOP model by 8.9% (0.00341 vs. 0.00338), and the AIO model by 26.9% (0.00341 vs. 0.00249). The rational lies in that the NJOP model deals with PFA planning according to the costs of modules that are estimated from historical product design data, which cannot take advantage of certain cost-effective modules through supply chain configuration. Consequently, optimal PFA planning can substantially benefit from joint optimization of supply chain configuration. Fig. 14 compares the supply chain costs of three models. While the AIO model results in the highest supplier, manufacturing, assembly and DC costs, the LFS marks the lowest cost figures of all supply chain entities. This echoes the importance of coordinated product family and supply chain decisions.

Table 20: Performance resulted from the bilevel model

Objective	Optimal value
Utility/Cost	34.1027×10^{-4}
Supplier Cost	2.7992×10^{7}
Manufacturing cost	2.2134×10^{7}
Assembly cost	2.3247×10^{7}
DCs cost	1.8296×10^{7}

Table 21: Performance resulted from the NJOP model

	J=3 R=2	J=3 R=3	J=3 R=4
Utility (10^4)	28.0289	28.0180	28.0191
$TC^S(10^7)$	3.8447	4.2103	4.5423
$TC^M(10^7)$	2.6123	2.9131	3.2135
TC^{AP} (10^7)	2.9411	3.1342	3.3132
$TC^D(10^7)$	1.8363	2.1134	2.3421
Utility /Cost (10^{-4})	24.9497	22.6481	20.8926

Table 22: Performance resulted from the AIO model

	J=2 R=2	J=2 R=3	J=2 R=4	J=3 R=2	J=3 R=3	J=3 R=4
Utility /Cost (10^{-4})	26.3273	27.7513	29.0326	32.7437	33.8123	33.0937
$TC^S(10^7)$	3.6712	3.4135	3.2613	3.0732	2.9032	3.1217
$TC^M(10^7)$	2.2321	2.8697	3.1043	2.8014	2.4038	2.5046

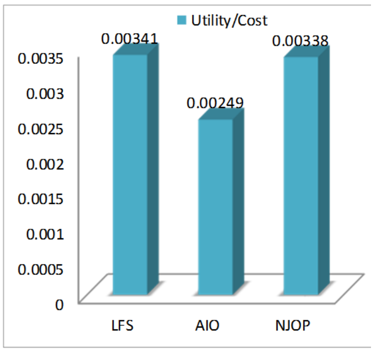

Fig. 13: Performance comparison for PFA planning

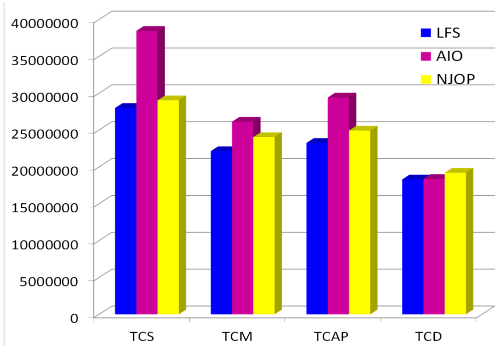

Fig. 14: Performance comparison for the supply chain

9. CONCLUDING REMARKS

Product family architecting deals with how to establish a PFA by optimal planning of compound modules and product variant configuration. It differs from product family configuration that is enacted within a given PFA. Recognizing the importance of coordinated product family and supply chain decisions, this paper proposes a joint optimization model for PFA planning with supply chain considerations. Decisions of supply chain configuration have a profound impact on not only the end cost of product families, but also the architecture of module configuration within a product family.

This paper reveals leader-follower Stackelberg game theoretic decisions underlying joint optimization of PFA planning and supply chain configuration. Product family architecting plays a leader's role for maximization of the customer-perceived utility per cost. Supply chain configuration acts as a follower that responds to the leader's decision regarding product variants, compound modules and base module instances, while providing feedback on the fulfillment costs to refine the leader's PFA decisions. Compound modules represent the common structures to be shared among product variants and selecting a combination of compound modules is consistent with product variant configuration, thus facilitating modeling of PFA planning with mathematical optimization models.

The power transformer case study verifies the reasonableness and superiority of the leader-follower bilevel optimization model. It indicates that bilevel programming with a leader-follower game excels in leveraging conflicting goals of competing optimization problems to arrive equilibrium solutions. The nested GAs are demonstrated to be an effective solution for this type of bilevel models.

The paper emphasizes formulation of the unique problem context and the case study is geared towards illustration of the research questions through conceptual findings. The reported work is limited to computational results and practical insights. In this regard, derivation of analytical results for the leader-follower joint optimization suggests itself to be an important avenue for future research. Since decision making in the real world usually involves multiple goals, extending the model to a multi-objective model with appropriate formulation would be one direction for future research. Further work can also focus on the computational efficiency of the bilevel model for large problems. More practical applications would shed light on validity of leader-follower bilevel optimization methods.

REFERENCES

Anandalingam, W. (1990). A solution method for the linear static Stackelberg problem using penalty functions. IEEE *Transactions on Automatic Control*, 35(10): 1170-1173.

Bard, J.F., Falk, J. (1982). An explicit solution to the multi-level programming problem, *Computers & Operations Research*, 9(1): 77-100.

Cao, J.L., Xiao, Y.Y, Xing, X.Y. (2011). A life cycle cost-based product design approach through integral architecture. *IEEE 18th International Conference on Industrial Engineering and Engineering Management.*

Cheng, L.C. (2011). Assessing performance of utilizing organizational modularity to manage supply chains: evidence in the US manufacturing sector. *International Journal Production Economics*, 131(2): 736-746.

Chiu, M.C., Okudan, G. (2011). An integrative methodology for product and supply chain design decisions at the product design stage. *Journal of Mechanical Design*, 133(2): 1-15.

Colson, B., Marcotte, P., Savard, G. (2005). A trust-region method for nonlinear bilevel programming: algorithm and computational experience. *Computational Optimization and Applications*, 30(3): 211-227.

Doran, D. (2003). Supply chain implications of modularization. *International Journal of Operations & Production Management*, 23(3): 316-326.

D'Souza, B., Simpson, T.W. (2003). A genetic algorithm based method for product family design optimization. *Engineering Optimization*, 35(1): 1-18.

Du, G., Yu, J.P., Wu, R.Y., et al. (2013). Leader-followers joint optimization of product family configuration and supply chain design. *International Journal of Knowledge, Innovation and Entrepreneurship,* 1(1-2): 6-23.

ElMaraghy, H.A., Mahmoudi, N. (2009). Concurrent design of product modules structure and global supply chain configurations. *International Journal of Computer Integrated Manufacturing,* 22(6): 483-493.

Fliege, J., Vicente, L.N. (2006). Multi-criteria approach to bilevel optimization. *Journal of optimization theory and applications*, 131(2): 209-225.

Fortuny-Amat, J., McCarl, B. (1981). A representation and economic interpretation of a two-level programming problem. *Journal of the Operational Research Society*, 32(9): 783–792.

Fujita, K., Amaya, H., Akai, R. (2013). Mathematical model for simultaneous design of module commonalization and supply chain configuration toward global product family. *Journal of Intelligent Manufacturing,* 24(5): 991-1004.

Fujita, K. (2002). Product variety optimization under modular architecture. *Computer-Aided Design*, 34(12): 953-

965.

Gibbons, R. (1992). Game theory for applied economists. Prince University Press: New Jersey.

Hernandez, G., Seepersad, C.C., Mistree, F. (2002). Designing for maintenance: A game theoretic approach. *Engineering Optimization*, 34(6): 561-577.

Huang, G.Q., Zhang, X.Y., Liang, L. (2005). Towards integrated optimal configuration of platform products, manufacturing processes and supply chains. *Journal of Operations Management*, 23(3-4): 267-290.

Jeroslow, R.G. (1985). The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, 32(2):146-164.

Jiao, J., Tseng, M.M. (1999). A methodology of developing product family architecture for mass customization. *Journal of Intelligent Manufacturing*, 10(1): 3-20.

Jiao, J., Tseng, M.M. (2000). Fundamentals of product family architecture. *Integrated Manufacturing Systems*, 11(7): 469-483.

Jiao, J., Simpson, T.W., Siddique, Z. (2007). Product family design and platform-based product development: A state-of-the-art review. *Journal of Intelligent Manufacturing,* 18(1): 5-29.

Jiao, J., Xu, Q., Wu, Z., Ng, N. (2009). Coordinating product, process, and supply chain decisions: A constraint satisfaction approach, *Engineering Applications of Artificial Intelligence*, 22 (7): 992-1004.

Jiao, J., Zhang, Y. (2005). Product portfolio planning with customer-engineering interaction. IIE *Transactions*, 37(9): 801-814.

Jiao, J., Tseng, M.M. (2013). On equilibrium solutions to joint optimization problems in engineering design. *CIRP Annals-Manufacturing Technology*, 62(1): 155-158.

Jiao, J., Tseng, M.M. (2004). Customizability analysis in design for mass customization. *Computer-Aided Design*, 36(8): 745–757.

Kaul, A., Rao, V. R. (1995). Research for product positioning and design decisions: An integrative review. *International Journal of Research in Marketing*, 12(4): 293–320.

Krishnendu, G.D. (2005). On Stackelberg games in a homogeneous product market. *European Economic Review*, 48: 549 -562.

Lai, Y. J. (1996). Hierarchical optimization: a satisfactory solution. *Fuzzy Sets and Systems*, 77(3), 321-335.

Lamothe, J., Hadj-Hamou, K., Aldanondo, M. (2006). An optimization model for selecting a product family and designing its supply chain. *European Journal of Operational Research*, 169: 1030-1047.

Lee, J., Chae, H., Kim, C.H., et al. (2009). Design of product ontology architecture for collaborative enterprises. *Expert Systems with Applications,* 36(2): 2300-2309.

Lewis, K., Chen, W., Schmidt, L. (2006). Decision making in engineering design. New York: ASME Press.

Li, L., Huang, G.Q. (2009). Multiobjective evolutionary optimization for adaptive product family design. *International Journal of Computer Integrated Manufacturing*, 22(4): 299-314.

Li, H., Wang, Y. (2008). Hybrid genetic algorithm for several classes of nonlinear bilevel programming problems. *Systems Engineering and Electronics*, 30(6): 1168-1172.

Liu, B. (1998). Stackelberg-Nash equilibrium for multilevel programming with multiple follows using genetic algorithms. *Computers and Mathematics with Applications*, 36(7): 79-89.

Mathieu, P.A. (1994). Genetic algorithm based approach to bilevel linear programming. *Research Operation*, 28(1):1-22.

Niwa, K., Nishizaki, I., Sakawa, M. (1998). Decentralized two-level 0-1 programming through genetic algorithms with double strings. *2nd International Conference on Knowledge-Base Intelligent Electronic Systems*: 278-284.

Oduguwa, V., Roy, R. (2002). Bilevel optimisation using genetic algorithm. *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems*: 322-327.

Oliveto, P.S., He, J., Yao, X. (2007). Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results. *International Journal of Automation and Computing*, 4(3): 281-293.

Pero, M., Abdelkafi, N., Sianesi, A., Blecker, T. (2010). A framework for the alignment of new product development and supply chains. *Supply Chain Management: An International Journal*, 15 (2): 115-128.

Qin, Y.H. (2012). A Stackelberg-game model in a two-stage supply chain. *Systems Engineering Procedia*, 3(2012): 268-274.

Rodriguez, K., Ashaab, A.A. (2005). Knowledge web-based system architecture for collaborative product development. *Computers in Industry*, 56(1):125-140.

Sakawa, M. and Nishizaki, I. (2012). Interactive fuzzy programming for multilevel programming problems: A review. *Int.J.Multicriteria Decision Making*, 2(3): 241-266.

Shahzad, M.K., Hadj-Hamou, K. (2013). Integrated supply chain and product family architecture under highly customized demand. J*ournal of Intelligent Manufacturing*, 24(5): 1005-1018.

Shabde, V.S., Hoo, K.A. (2008). Optimum controller design for a spray drying process. *Control Engineering Practice*, 16(5): 541-552.

Stone, R.B., McAdams, D.A., Kayyalethekkel, V.J. (2004). A product architecture-based conceptual DFA technique. *Design Studies*, 25(3): 301-325.

Truong, T.H., Azadivar, F. (2005). Optimal design methodologies for configuration of supply chains, International Journal of Production Research, 43(11): 2217-2236.

Ulku, S., Schmidt, G.M. (2011). Matching product architecture and supply chain configuration. Production and Operations Management, 20(1): 16-31.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, 24(3): 419-440.

Verdouw, C. N., Beulens, A. J. M., Trienekens, J.H., Verwaart, T. (2010). Mastering demand and supply uncertainty with combined product and process configuration. *International Journal of Computer Integrated Manufacturing*, 23(6): 515-528.

Von, S. H. (1952). The theory of the market economy. Oxford University Press: Oxford.

Yang, D., Jiao, R.J., Ji, Y., Du, G., Helo, P., Valente, A. (2015) Joint optimization for coordinated configuration of product families and supply chains by a leader-follower Stackelberg game. *European Journal of Operational Research*, 246(1): 263-280.

Zhang, X., Huang, G.Q., Rungtusanatham, M.J. (2008). Simultaneous configuration of platform products and manufacturing supply chains. *International Journal of Production Research*, 46(21): 6137-6162.

Zhu, H.P., Liu, F., Shao, X.Y., Zhang, G.J. (2010). Integration of rough set and neural network ensemble to predict the configuration performance of a modular product family. *International Journal of Production Research*, 48(24): 7371-7393.