Greedy heuristics for determining a product family bill of materials

Radwan El Hadj Khalaf, Bruno Agard, Bernard Penz

GREEDY HEURISTICS FOR DETERMINING A PRODUCT FAMILY BILL OF MATERIALS

Radwan El Hadj Khalaf
Grenoble INP -- CNRS -- UJF
France
radwan.el-hadj-khalaf@g-scop.inpg.fr

Bruno Agard
Département de Mathématiques et Génie Industriel
Ecole Polytechnique de Montréal
Canada
bruno.agard@polymtl.ca

Bernard Penz
Grenoble INP -- CNRS -- UJF
France
bernard.penz@g-scop.inpg.fr

ABSTRACT

When designing a new product family, designers and manufacturers have to define simultaneously the product structure and its supply chain. This leads to a complex optimization problem to solve in order to satisfy diversified customers' requirements with various options and variants. The paper focuses on the first step of this design problem. It consists in selecting a set of modules that will be manufactured in distant facilities and shipped in a nearby location plant for a final assembly operation under time limits. The objective is to determine the set of modules able to define the bill of materials of each finished product in order to minimize assembly costs. We propose in this paper two heuristic strategies to solve the problem. We provide experiments on small instances which we compare with optimal solutions and we provide also experiments on big instances to compare performance of each heuristic.

KEYWORDS

Product family design, bill of materials, integer programming, greedy heuristic, assembly costs

1. INTRODUCTION

Nowadays, the growing demand for customized products involves an increasing number of product variants and options. It follows a complex product diversity to manage. This variety must be controlled in term of product, process and supply chain costs, as well as customer lead-time. Consequently, when designing a new product family, a consistent approach is necessary to quickly define a set of variants and the relevant supply chain, in order to guarantee the customers' satisfaction and to minimize the total investment and operating cost of the global supply chain (Lamothe, J. et al. 2006).

A product family is composed by similar products that differ by some characteristics such as options. For example, the basic car model may contain few options in order to minimize the sale price. Then, some options can be added to this basic model like air-conditioning, automatic gear box or diesel engines and so on.

There are two extreme production strategies that a company can use. The first one consists to make to stock the different products. This leads to select a minimum set of standardized products (Briant, O., Naddef, D., 2004), that could include supplementary options to meet diversified customer requirements. However, storage costs may be too high because of the large product portfolio. The second strategy consists to produce only when an order is received. In this case, the lead time may be higher leading to the non satisfaction of the customer. An intermediate strategy consists to manufacture pre-assembly components, called modules, for stock and to assemble them when an order is planed. The advantage of such strategy is to reduce the lead time and to avoid great storage costs.

In this paper, we explore this production policy where modules are manufactured in distant location facilities for cost minimization. Those modules are shipped and assembled in a nearby location facility in order to ensure a short lead-time for the customers.

We present two heuristic strategies to define the bill of materials of each finished products: (1) the first one consists on exploring the finished product set and determining the most suitable bill of materials for each one and (2) the second strategy consists on exploring the module set, selecting the most interesting one and inserting it in the bill of materials of compatible finished products.

This work is of a great utility for us afterwards, because we need to dtermine quickly the product bills of materials in order to affect the resulting modules on the distant location facilities.

Literature proposes various approaches for the design of product families. Some product design methodologies focus on the product architecture (Dahmus, J. B., et al., 2001), (Jiao, J., Tseng, M., 1999), this is advantageously supported with modular design (Hung, C. C., Kusiak, A., 1998), component / product / process standardization (Kota, S., et al. 2000), (Lee, H. L., Tang, C. S., 1997), (Thonemann, U. W., Brandeau, M., 2000). Different methodologies concentrate on process standardization (Lee, H. L., Tang, C. S., 1997), process resequencing (Lee, H. L., 1996) or generic assembly routing (Gupta, S., Krishnan, V., 1998), (He, D. W., Kusiak, A., 1997). From another point of view, authors concentrate on supply chain design methodologies, by integrating customers and suppliers in uncertain environment (Davis, E. W., 1992), centred on stocks management (Lee, H. L., Billington, C., 1992) and for distant facility location (Van Roy, T., 1986).

In all these works, design of product, process and supply chain are integrated two by two. However, there is some recent works dealing with a global design modeling. Agard, B., et al. (2006) propose a genetic algorithm to minimize the mean of finished product assembly times for a given demand. Agard, B., Penz, B., (2007) propose a model for minimizing module production costs and a solving approach based simulated annealing. Lamothe, J., et al. (2006) use a generic bill of material representation in order to identify simultaneously the best bill of material for each product and the optimal structure of the associated supply chain

In section 2 we give a more detailed description of the problem and we propose an Integer Linear Program model. Section 3 is devoted then to the description of the two heuristic strategies. Some computational experiments are given and analyzed in section 4. Finally concluding remarks and perspectives are proposed in section 5.

2. PROBLEM PRESENTATION

Consider the following industrial context: a producer receives customers' orders for finished products containing options and variants. Each individual product is then manufactured from a set of modules that come from various suppliers (El Hadj Khalaf, R., et al. 2008).

Consider now that the producer has only a short delay (T) to respond to customer demands. This delay is less than the necessary time to assemble products from elementary components. In addition to this, the producer has to provide the product exactly like the customer demand (without extra options). This constraint comes from technical considerations or simply to avoid supplementary costs.

To satisfy customers, the producer brings pre-assembled components, called modules, from many suppliers which are located in distant facilities around the world. The suppliers' facilities are characterized by a very weak production costs. Then, the modules are assembled in the producer facility which we assume to be very close to the customers and thus characterized by its great reactivity and a reduced lead-time. Our problem consists then on defining the bill of materials of each finished product in order to minimize the total assembly costs.

Specifying the problem assumptions: a product or a module is considered as the set of functions that it must fill, then:

- a function F_k is a requirement that must be ensured by the finished product.
- a module M_j is an assembly of functions that could be added with other modules to make a finished product.
- a finished product P_i is an assembly of modules that corresponds exactly to at least one customer demand.

Let introduce the following notations:

- $F = \{F_1,...,F_q\}$: set of q functions that can appeared in both finished products and modules.
- $P = \{P_1,...,P_n\}$: set of n possible finished products that may be demanded by at least one customer. We note D_i the estimated

demand of the product P_i during the life cycle of the product family.

- $M = \{M_1,...,M_m\}$: set of m possible modules.
- CF_j: the management fixed cost of module M_j in the nearby facility.
- CV_j: the assembly variable cost of module M_j in the nearby facility.
- w_j : the necessary time to assemble the module M_j in a finished product (which is fixed to 1 for all modules).
- T: the available time to assemble a finished product from modules.
- WhP_i, WhM_j: the weights of respectively product P_i and module M_j (which represent the number of existing functions in a product or a module).

Under these assumptions, we can represent a product (or a module) by a binary vector of size q. Each element shows whether the corresponding function is required in the product (value = 1) or not (value = 0).

The set M contains m modules. It may be a selection of modules defined by the engineering or all the possible modules issued from the whole combinatory.

The problem is now to determine the subset $M' \in M$, of minimum cost, such that all products in P can be built in a constrained time window T with elements from M'. Concerning the products, the goal is to determine which bill of material is the most suitable (Figure 1).

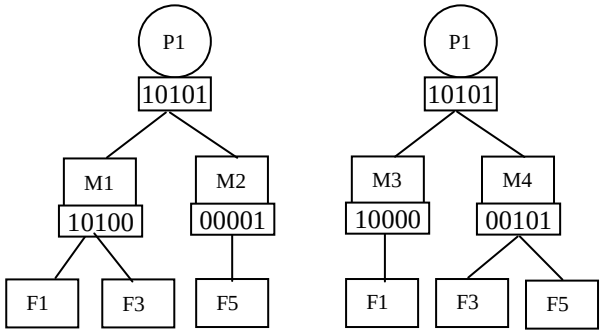

Figure 1 alternative bills of material

We have done a model of the problem using an Integer Linear Program formulation. Our objective consists in minimizing costs linked to the producer activity. These costs are: fixed costs due to modules management in the nearby location facility and modules assembly costs in the nearby location facility.

$$Z = \min \sum_{j=1}^{m} CF_j Y_j + \sum_{j=1}^{m} CV_j \left(\sum_{i=1}^{n} D_i X_{ij}\right)$$

s.t.

$$AXi = Pi \qquad \forall i \in \{1,..., n\} \qquad (1)$$

$$\sum_{j=1}^{m} w_j X_{ij} \le T \qquad \forall i \in \{1,..., n\} \qquad (2)$$

$$X_{ij} \le Y_j \qquad \forall i \in \{1,..., n\} \forall j \in \{1,..., m\} \qquad (3)$$

$$Y_j, X_{ij} \in \{0,1\} \qquad \forall i \in \{1,..., n\} \forall j \in \{1,..., m\} \qquad (4)$$

Where X_{ij} = 1 if module M_j is used in the bill of materials of product P_i, 0 otherwise. Y_j = 1 if module M_j is selected (belongs to M'), 0 otherwise. A is the binary matrix whose the column j is the vector M_j. X_j is the column vector composed by the variables X_{ij}.

The objective function minimizes the costs occurring in the nearby location facility, where $\left(\sum_{i=1}^{n} D_i X_{ij}\right)$ is the total need of module M_j.

Constraint (1) shows that a finished product P_i must be assembled exactly like the customer demand. Constraint (2) indicates that products must be assembled within the time window T in order to respect the delivery time. Constraint (3) traduces the relation between X_{ij} and Y_j variables. If a module is used in the bill of materials of some products then it belongs to M'.

The problem described here contains the set partitioning problem. We then conclude that it is NP-hard in the strong sense.

3. HEURISTIC DESCRIPTION

3.1. The product building heuristic (PBH)

The main idea of this heuristic is to affect an attractive coefficient to each module and to determine the bill of materials of products one by one (by using Cplex) in such a way as to maximize the sum of attractive coefficients of the product's components. So at each iteration, we resolve the problem described above but only for one finished product.

The coefficient expression is given as follow:

$Coef_j = F1(CF_j) + F2(CV_j) + F3(WhM_j)$ where:

$$F1(CF_j) = 100\ \frac{\overline{CF}}{CF_j} \text{ with } \overline{CF} = \sum_{j=1}^{m} \frac{CF_j}{m}$$

$$F2(CVj) = 100\ \frac{CV_j \sum_{i \leftrightarrow j} D_i}{\overline{CV} \sum_{i=1}^{n} D_i} \text{ with } \overline{CV} = \sum_{j=1}^{m} \frac{CV_j}{m}$$

and $i \leftrightarrow j$ means that products P_i must be compatible with module M_j (M_j haven't extra functions than P_i). So that the term $\sum_{i \leftrightarrow j} D_i$ represents the needs of module M_j if we use it in the remaining compatible products at iteration *k*.

As much as a module is compatible with finished products *F2* becomes bigger. So that, this coefficient favours modules which are compatible with many finished products.

We have used two different functions F3:

$$F3_1(WhM_j) = 100 \text{ if } WhM_j \leq \left\lceil \frac{q}{T} \right\rceil, 0 \text{ otherwise.}$$

This function favours small modules which is necessary when cost configuration is such that total fixed costs are greater than total variable costs. In such case it is better to select small modules because they are compatible with much more products, so using them leads to a solution with small number of modules which reduces fixed costs which represent the great part of objective function (el hadj khalaf, R., et al. 2008).

$F3_2(WhM_j) = 100$ if $WhM_j \geq \overline{WhP}$, 0 otherwise.

With $\overline{WhP} = \sum_{i=1}^{n} \frac{WhP_i}{n}$.

This function favours relatively big modules which is necessary when cost configuration is such that total variable costs are greater than total fixed costs. In such case using big modules allows to have small requirements and so to reduce total variable costs which represent the great part of the objective function.

Since we don't know in advance the report between fixed and variable costs, we test this heuristic with both functions F3. Moreover, to improve this heuristic we sort finished products by increasing (and decreasing) order of their weights.

3.2. The module selecting heuristic (MSH)

This heuristic is quite simple; the idea is to select the module having the smallest value of $CF_j + CV_j \sum_{i \leftrightarrow j} D_i$ and insert it in the bill of materials of compatible finished products.

To improve this heuristic also, we determine first at each iteration *k* the ideal weight of the module to be selected by the formula: $Wh_k = \left\lceil \frac{1}{n_k} \sum_i \frac{WhP_{ik}}{W_{ik}} \right\rceil$

with: WhP_{ik} is the weight of the remaining functions of product P_i at iteration *k* which are not covered yet and $W_{ik} = T - W0_{ik}$ where $W0_{ik}$ is the number of modules inserted in the bill of materials of the product P_i up to the iteration *k*, n_k is the number of remaining products at iteration *k* (those which haven't the complete bill of materials yet).

This operation aims to avoid selecting unitary modules (whose weight is 1) at the beginning which is proved to be bad for the solution quality.

So we can summarize this heuristic as follow at iteration *k*:

1. Calculate the ideal weight Wh_k of the module to be selected. With $WhP_{i1} = WhP_i$ and $n_1 = n$ and $W_{i1} = T$ and $P_{i1} = P_i$
2. Select the module M_j whose weight is equal to Wh_k and having the smallest value of $CF_j + CV_j \sum_{i \leftrightarrow j} D_i$. (i represents the products P_{ik} compatible with M_j).
3. Insert the module selected in the bill of materials of compatible finished products.
4. For these compatible products update the product code: $P_{ik} = P_{i(k-1)} - M_j$
5. Update also $W_{ik} = W_{i(k-1)} - 1$ which represents the maximum number of modules that can be inserted in product P_{ik}.
6. For products having $W_{ik} = 1$, complete their bill of materials by the alone compatible module.
7. Repeat these steps until constructing all bills of materials.

4. COMPUTATIONAL EXPERIMENTS

4.1. Data sets and experimental conditions

We have randomly generated ten small instances for five different sizes; $q \in \{8, 10, 11, 12, 13\}$ on which the module set, the finished product set, the demand D_i, the assembly operating times w_j, the minimum function number per product *Minf* and the maximum function number per product *Maxf* were fixed. Table (1) summarizes the different parameters for each instance size.

q	**8**	**10**	**11**	**12**	**13**
m	255	1023	2047	4095	8191
n	30	40	60	80	100
Minf	3	3	3	4	4
Maxf	6	7	8	8	9

Table 1 Instance parameters

The demand D_i of a product P_i is a decreasing function on the product function number. So that, as soon as the finished product contains more options, then its demand becomes lesser than if it had a few ones. The assembly operating times w_j were fixed to 1, so that the constraint (2) turns out on a limitation of the number of modules in a bill of materials.

In order to simplify the problem data, we introduced some rules on the different costs:

- $CF_j = \alpha\left(f(q_j) + \lambda 1_j\right)$
- $CV_j = \beta\left(f(q_j) + \lambda 2_j\right)$

Where q_j is the number of existing functions in module M_j, f is the square root function $f(q_j) = \sqrt{q_j}$, we estimate that this function represents at best the relation between costs and q_j than the identity and the square functions (El Hadj Khalaf, R., et al. 2008). α, β are coefficients used to scan different cost configurations. λ1, λ2 are jamming factors.

For each instance size, three cost files are generated by the method described above. The aim is to scan different cases of report between fixed and variable costs.

Table 2 gives the values affected to α and β for each cost. For cost 1 fixed costs are prependerant than variable ones while for cost 2 the two costs are almost equivalen, for cost 3 variable costs are the most preponderant.

Cost	**1**	**2**	**3**
α	1000	240	100
β	0.10	0.40	0.50

Table 2 Cost Configurations

We have $8\% \le \lambda 1, \lambda 2 \le 12\%$. For the tests, T was varied from *Minf* to *Maxf*. Heuristics are then tested on all instances and all costs. Finally, the mean value of the ten instance objective value is recorded for comparison analyses.

4.2. Result analyses

At first, our objective is to compare heuristic results with optimal solutions. This is why we have tested them on an instance of size eight. Table 3 shows the gap rate between the both heuristic results and the optimal solutions for the three cost tests.

T		**3**	**4**	**5**	**6**
Cost 1	**PBH**	54%	28%	8%	0%
	MSH	29%	9%	11.5%	0%
Cost 2	**PBH**	35%	19%	7%	0%
	MSH	15%	2.2%	2.4%	0%
Cost 3	**PBH**	25%	16%	12%	7%
	MSH	9%	2.8%	4.3%	7%

Table 3 Gap rate between heuristic results and optimal solutions for $q = 8$

The first lecture of these results shows that the MS heuristic is better than the PB one. In addition to this, we note that both heuristics run well for medium and high values of T and specially for cost 2 and cost 3 thus when assembly fixed costs are not greater than assembly variable costs. This leads to think that these heuristics optimize well variable costs.

Our second objective is to test the two heuristics on relatively big instances in order to compare there performance for a higher problem sizes. That was the object of tests on sizes 10, 11, 12 and 13.

Figures 2, 3 and 4 show the gap rate (as a percentage) between the results of the product building heuristic and the module selecting heuristic (MSH objective value is considered as the reference value).

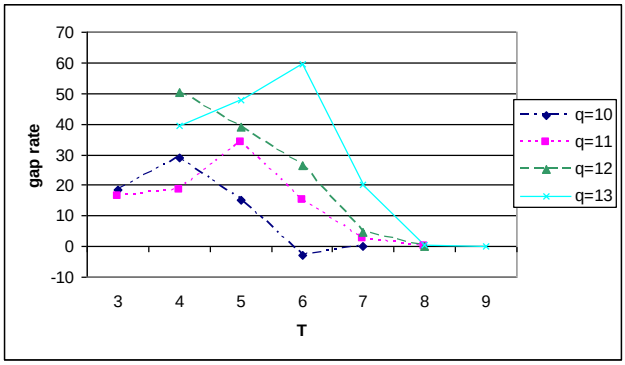

Figure 2 Gap rate between PBH & MSH for cost 1

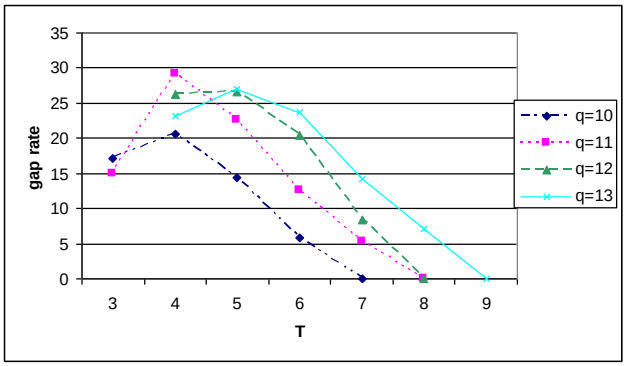

Figure 4 Gap rate between PBH & MSH for cost 2

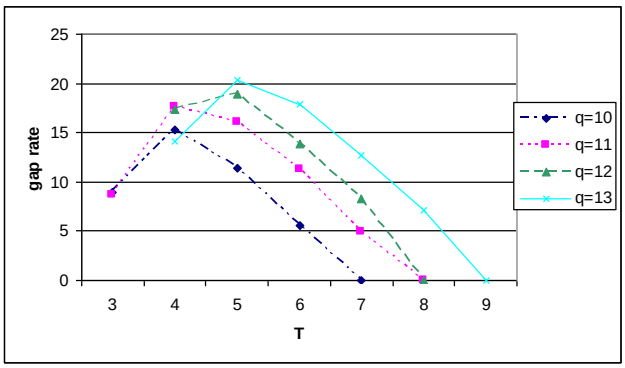

Figure 5 Gap rate between PBH & MSH for cost 3

These figures confirm the results obtained for *q=8*, MSH is much better than PBH and this is true for practically all sizes, delays, and costs. We can note also that the gap rate increases as the problem size increases too. So the gap for *q=13* is generally greater than the one for *q=12* and so on.

We note also that generally the gap rates are more important for cost 1 than cost 2 and for cost 2 than cost 3. This indicates that MSH optimizes fixed costs more effectively than PBH.

Finally Figure 5 shows the computational time of PBH for *q=13*. This figure indicates that PBH is very consuming in computational time than MSH which has an average time of one second per instance for all sizes (very quick). While PBH needs more and more computational time as soon as the problem size increases (an average of 200 seconds for *q=12*). This is expected because (1) PBH tests two indicators with two sorting methods (four combination) and especially (2) it uses Cplex to determines the product bill of materials which necessitates much more time because we treat a difficult problem.

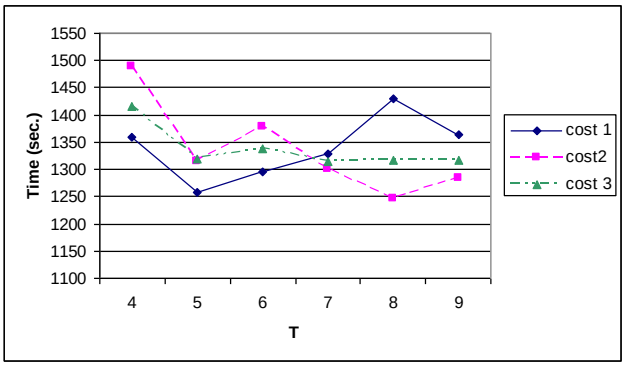

Figure 3 Computational time of PBH for q = 13

At the end, we present table 4 which show the gap rate (as a percentage) between the MSH results and the problem linear relaxation results for each problem size, cost file and delivery delay.

		T						
q	**Cost**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
10	**1**	130	65	28	17	0		
	2	65	24	8	4	1.4		
	3	36	16.7	12.8	15.7	19.4		
11	**1**	197	105	39	19	11	0	
	2	89	33.7	12	6.4	5.9	6	
	3	47	22.6	16.3	17.9	22.7	26.9	
12	**1**		141	73	24.4	15.1	0	
	2		44.8	18.4	9.3	9.3	11.5	
	3		28.5	19.8	21	25.4	33.8	
13	**1**		202	110	31.3	20.4	16.5	0
	2		64.2	27.2	11.8	10.3	11.2	15.1
	3		36.1	23.4	22.2	25.8	30.6	38.8

Table 4 Gap rate between MSH results and linear relaxation results

As we can see, the gap is quite reasonable for medium and big delays, it could reaches till 0% in some cases (when $T = Maxf$). However, it is very great for small values of T because in this case the problem itself becomes much more difficult and even for small instances Cplex spends much more time to reaches the optimal.

We note again, that the gap rate is more important (for small delays) for the cost 1 contrarily to the other costs. We can hope that MSH provides relatively good solutions for cost configurations on which assembly fixed costs are not preponderant than assembly variable costs.

5. CONCLUSION

This paper was dedicated to a difficult industrial problem arising when companies try to offer a large variety of products to consumers. In this problem, a choice of components (modules) has to be efficient. These modules are produced for stock, and used in the last stage, in the assembly line. Several authors considered this problem, using different assumptions - a function can appear twice in a final product, a final product can be substituted by another one containing more functions - but few papers consider the problem in which each final product must correspond exactly to the demand.

We have focus on the assembly operation and we tried to determine an efficient module set allowing to assemble products and to avoid function redundancy while respecting the delivery delay. The objective function consists in optimizing the costs occurring in the producer activity.

Our aim was to analyze to strategies of heuristic design. The first one consists in determining the bill of materials of finished products one by one by fixing attractive coefficients for each module. The second strategy takes the ideal module verifying the selecting criterion and inserts it in the bill of materials of compatible finished products.

Besides the computational time speed, our tests reveal that the MSH strategy is by far the one which gives the best results. This can be explained by the difficulty to find good module coefficients and especially the problem structure in which solutions must take into account the interactions between finished products which make very difficult the determining of one product bill of materials independently from the other ones.

Finally, the MSH results encourage us to use it as an initial solution for a metaheuristic resolution in the future. It would be also interesting to extrapolate this heuristic in order to take into account the logistic chain phase.

REFERENCES

Agard, B., Cheung, B., and da Cunha, C., (2006), " Selection of a module stock composition using genetic algorithm", in 12th IFAC Symposium on Information Control Problems in Manufacturing – INCOM, Saint-Etienne.

Agard, B., Penz, B., (2007), "A simulated annealing method based on a clustering approach to determine bills of materials for a large product family", submitted to: International Journal of Production Economics.

Briant, O., Naddef, D., (2004), "The optimal diversity management problem", Operations Research, Vol 52-4, pp. 515-526.

Dahmus, J. B., Gonzalez-Zugasti, J. P., and Otto, K., (2001), "Modular product architecture", Design studies, Vol 22, pp 409-424.

Davis, E. W., (1992), "Global outsourcing: have U.S. managers thrown the baby out with the bath water?", Business Horizons, pp 58-65.

Gupta, S., Krishnan, V.,(1998), “Product family-based assembly sequence design methodology”, IIE Transactions, Vol. 30, pp 933-945.

El Hadj Khalaf, R., Agard, B., Penz, B., (2008), “Product and supply chain design using a two-phases optimization approach”, International Conference on Information Systems, Logistics and Supply Chain, Madison.

He, D. W., Kusiak, A., (1997), “Design of assembly systems for modular products”, IEEE Transactions on Robotics and Automation, Vol. 13-5, pp. 646-655.

Hung, C. C., Kusiak, A., (1998), “Modularity in design of products and systems”, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 28-1, pp. 66-77.

Jiao, J., Tseng, M., (1999), “A methodology of developing product family architecture for mass customization”, Journal of Intelligent Manufacturing, Vol. 10, pp. 3-20.

Kota, S., Sethuraman, K., and Miller, R., (2000), “A metric for evaluating design commonality in product families”, Journal of Mechanical Design, Vol. 122, pp. 403-410.

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, Vol. 169, pp. 1030-1047.

Lee, H. L., (1995), “Product universality and design for supply chain management”, Production Planning and Control, Vol. 6-3, pp. 270-277.

Lee, H. L., (1996), “Effective inventory and service management through product and process redesign”, Operations Research, Vol. 44-1, pp. 151-159.

Lee, H. L., Billington, C., (1992), “Managing supply chain inventory: Pitfalls and opportunities”, Sloan Management Review, Vol. 33-3, pp. 65-73.

Lee, H. L., Tang, C. S., (1997), “Modelling the costs and benefits of delayed product differentiation”, Management Science, Vol. 43-1, pp. 40-53.

Van Roy, T., (1986), “Cross decomposition algorithm for capacity facility location”, Operations Research, Vol. 34, pp. 145-163.

Pine B. J., (1993), “II. Mass Customization: The new Frontier in Business Competition”, Harvard Business School Press, Boston.

Thonemann, U. W., Brandeau, M. (2000), “Optimal commonality in component design”, Operations Research, Vol. 48-1, pp. 1-19.