J Intell Manuf (2010) 21:703–716
DOI 10.1007/s10845-009-0247-0

An experimental study for the selection of modules and facilities in a mass customization context

Radwan El Hadj Khalaf · Bruno Agard · Bernard Penz

Received: 17 July 2008 / Accepted: 5 February 2009 / Published online: 22 February 2009

Abstract To design an efficient product family, designers have to anticipate the production process and, more generally, the supply chain costs. But this is a difficult problem, and designers often propose a solution which is subsequently evaluated in terms of logistical costs. This paper presents a design problem in which the product and the supply chain design are considered at the same time. It consists in selecting a set of modules that will be manufactured at distant facilities and then shipped to a plant close to the market for final, customized assembly under time constraints. The goal is to obtain the bill of materials for all the items in the product family, each of which is made up of a set of modules, and specifying the location where these modules will be built, in order to minimize the total production costs for the supply chain. The objective of the study is to analyze both, for small instances, the impact of the costs (fixed and variable) on the optimal solutions, and to compare an integrated approach minimizing the total cost in one model with a two-phases approach in which the decisions relating to the design of the products and the allocation of modules to distant sites are made separately.

Keywords Modular design · Product family design · Supply chain design · Mass customization · Bill of materials

R. E. H. Khalaf · B. Penz
Laboratoire des Sciences pour la Conception, l′Optimisation et la Production (G-SCOP), Grenoble INP-CNRS-UJF, 46 Avenue Félix-Viallet, 38031 Grenoble Cedex 1, France

B. Agard (✉)
Département de Mathématiques et Génie Industriel, Centre Interuniversitaire de Recherche sur les Réseaux d′Entreprise, la Logistique et le Transport (CIRRELT), École Polytechnique de Montréal, C.P. 6079, Succ. Centre-Ville, Montreal, QC H3C 3A7, Canada
e-mail: bruno.agard@polymtl.ca

Introduction

In recent years, and for a number of reasons, industrial markets have changed. First, globalization is leading to an opening up of those markets. This provides customers with greater choice, which enables them not only able to compare prices, but to find products which correspond exactly to their requirements (Xu et al. 2007).

Today, the growing demand for customized products involves an increasing number of product variants and options, which results in the need to manage complex product diversity. Such variety must be controlled, in terms of product, process, and supply chain costs, as well as customer lead-time. In order to provide an efficient solution to this problem without extensive product proliferation, companies may focus on "mass customization". Mass customization deals with large product portfolios, flexible manufacturing systems, and extended supply chains (Pine II 1993).

Under the pressure of competition, the whole process of supply, warehousing, production, and transportation has been studied. Logistics, which played a minor role in the past, plays a decisive one in today's strategies. In the attempt to satisfy demand, the reliability and punctuality of deliveries form an essential part of that logistics. Along with flexibility in production and deliveries, however, costs must be optimized (Zhiuong and Fanqhua 2007).

In this context, a new design strategy is developing. Product family design must now take into account not only product diversity, but also definition of the process and the supply chain (Yang et al. 2007). A consistent approach to product family design is needed in order to guarantee customer satisfaction, as well as to minimize the total investment on the part of producers in the product and in the operating cost of the global supply chain (Lamothe et al. 2006).

Through increasing competition and the necessity to reduce costs, producers are forced to integrate the production chain of their suppliers. This is now possible because of well-developed computer technologies and new communication technologies. An example of these highly merged production chains of producers and suppliers can be found in the automotive industry.

A challenge for product family design is to control the number of sub-elements with a view to maintaining the storage cost of components at a reasonable level. Modularity is a good way to achieve such a compromise, as suppliers put together modules containing combinations of functionalities needed in the finished products. As a result, the producer uses a limited number of modules to assemble a product, and one module can be used in many products (Hale and Kusiak 1998).

The strategy of working with modules has the advantage of reducing the final assembly time and the number of elements used in the final assembly phase. As a consequence, organization, storage, and transportation are all simplified (Da Cunha et al. 2007). The product family provides the ideal support for this approach.

A product family is composed of similar products which differ in some characteristics such as options. For example, the basic car model may offer few options, in order to minimize the retail price. Then, based on individual customer requests, options can be added to this model, like air-conditioning, an automatic gear box, or a diesel engine, and so on.

There are two extreme production strategies which a company can use. The first consists of manufacturing the various products to stock. "Make-to-stock" may lead to high storage costs if too many alternative versions have to be considered. Such a situation would, in turn, lead to the selection of a minimum set of standardized products (Briant and Naddef 2004), and could include supplementary options to meet diversified customer requirements. However, standardization costs may also be too high, if many unnecessary functions are offered to customers. The second strategy consists of producing only when an order is received. In this case, the lead time may be longer, leading to a failure to satisfy the customer. An intermediate strategy would be to manufacture preassembly components, called modules, for stock, and to assemble them when an order is received. The advantages of this strategy are that the lead time can be reduced, and high storage and standardization costs can be avoided.

For the selection of modules that permit the manufacture of mass customized products under time constraints, Agard et al. (2006) propose a genetic algorithm to minimize the mean final assembly time for a given demand, and Agard and Penz (2009) propose a model for minimizing module production costs based on simulated annealing. However, because these models do not consider the number of modules to be manufactured, it is not possible to consider limitations in production capacities for the different production facilities, nor any variable costs that are more representative of the real industrial context. Lamothe et al. (2006) use a generic bill of materials representation to identify the best bill of materials for each product and the optimal structure of the associated supply chain simultaneously, although this approach requires that a predefined generic bill of materials be generated for the product family.

The problem of the assignment of modules to distant location facilities is very close to the classical facility location problem. The purpose of facility location models is to select a set of facilities among potential alternatives to serve the needs of customers while minimizing investment, production, and distribution costs to the supplier, whereas the module assignment problem consists in determining at which facility each module should be produced in order to minimize costs. In the literature, a wide variety of location models have been proposed. Hale (2005) provides an extensive bibliography devoted to facility location, and good surveys of past research can be found in Daskin (1995). The main difference between this problem and the module assignment problem is that the facility location problem treats a demand of one kind of product in general, while in the module assignment problem, various kinds of modules are produced at the distant facilities. There is, however, some recent work dealing with the k-product facility location problem. Huei-Chuen and Rongheng (2008) present an approximation algorithm for this problem, and show that it provides an optimal solution in a specific cost structure. This problem is similar to the one described here, except that it does not consider information about the demand for each product, and the production facilities do not have limited capacity.

It is not possible to know the exact demand Di for each final product when the diversity of that product is too great. In the car industry, for example, the number of different cars can be as large as 1,418,701,950,016 for a particular model (Renault Megane) (Amilhastre et al. 2002). However, some techniques make it possible to estimate this demand from the consumption of each component, which, by contrast, are known with good accuracy from historical data (da Cunha 2004). These techniques are proved to be very efficient. Historical data gives the total demand D of all products (all options confused) and the demand of each component independently from the other ones. To estimate the demand of an specific finished product (which is a combination of components), we have to solve an optimization problem on which the objective function is the maximization of the total product entropy value. The entropy is a function that measures the disorder of a system. In our case, the entropy of a product (P_i) is equal to $\mu(P_i) \cdot log\,(\mu(P_i))$ where $\mu(P_i) = D_i/D$ is the probability of the demand of P_i. D_i is the demand of P_i (that has to be determined) and D is the total demand of all products (which is estimated from historical sale data). Jaynes (1957, 1982) uses the entropy notions and shows that it could give a good and unique estimation of event probabilities in the case of an incomplete information. As constraints, it is obvious that the sum of the product demands must be equal to the total demand D, and the sum of the demand of products containing a component (a_j) must be equal to the demand of (a_j). This formulation remains valid if we have got information data about sales of any combination of components (we just have to integrate this information in the constraints). Uzawa algorithm (Bacuta 2007) is an efficient technique that gives a good approximated solution of the above problem. It is a gradient algorithm of a constant pace.

In this paper, we explore the production policy according to which modules are manufactured at distant facilities for cost minimization purposes. Those modules are shipped and assembled at a nearby facility in order to ensure a short lead-time for the customer. The electric beam family of products, largely used in the car industry, is an example of this (Lamothe et al. 2006). We compare two modeling strategies: (1) a two-phase approach, often followed in industrial contexts, in which the costs associated with production at the nearby and distant facilities are optimized separately; and (2) an integrated approach, in which the process and the supply chain costs are taken into account simultaneously. The aim of this paper is to give a detailed analysis of the optimal solutions to each approach by scanning many cost configurations. Small instances are used here to obtain optimal solutions. We then focus on the advantages and drawbacks of the two approaches, in particular comparing their solution quality and computational time.

The methods proposed in this paper are intended to be used as tools to help in the strategic decision making associated with designing a product family (design of the bills of materials and initial selection of production facilities). The objective of the paper is to both understand the phenomena on small size problems (where it is possible to arrive at an optimal solution) and make it possible to compute large problems. The two-phase approach is dedicated to computing large problems.

A detailed description of the problem is provided in section "Mathematical modeling". Notations are explained in section "Notations", and then Mixed Integer Linear Program models are given in sections "A two-phase modeling approach" and "An integrated modeling approach" for the two-phase and integrated approaches. Some computational experiments are given and analyzed in section "Computational experiments". Finally concluding remarks and perspectives are proposed in section "Conclusion".

Mathematical modeling

Consider the following industrial context (Fig. 1). The producer receives customers' orders for finished products containing options and variants. Each individual product is then manufactured from modules provided by various suppliers.

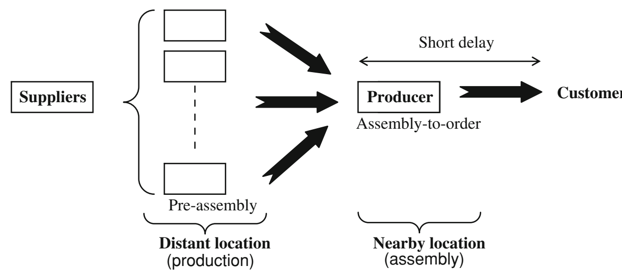


Fig. 1 Structure of the supply chain

The producer has only a short time (T) in which he is expected to respond to a customer's order, and it corresponds to the period between receipt of the customer's order and the moment when the component/product is required. When this period is shorter than the time needed to assemble the products from elementary components, the supplier needs to manufacture some preassembled elements (modules) in advance, which will save time during final assembly. When T is short, the supplier needs to react more quickly. In addition, the producer has to provide the product precisely according to the customer's requirements (no extras). This constraint derives either from technical considerations or simply from a desire to avoid the supplementary cost of offering none requested options. Our study also addresses the case where T is long enough not to be a constraint.

To satisfy customer orders, the producer brings in preassembled components, called modules, from many suppliers located at facilities around the world. The production costs incurred by these suppliers are low. Then, the modules are assembled at the producer's facility, which, we assume, is close to the customers, and thus characterized by a rapid reaction time and a short lead-time.

The strategic problem is, then, to design the product family, i.e. to determine the bill of materials for each product. A product will be made up of a set of modules. For modules which appear on at least one bill of materials, we have to determine where those modules must be produced in order to minimize production and transportation costs. Fixed costs represent the costs for starting the production (buy, update, set up machines, etc.). Variable costs are proportional to the number of each module to be manufactured, which is linked to the definition of the bill of materials. Variable costs represent production costs, storage costs, etc. The goal of the paper is to understand how problem-solving evolves as a function of the relative importance of fixed and variable costs.

Notations

A product (or a module) is considered as the set of functions that it contains. It is currently modeled with a binary vector in which 1 means that the function is present in the product (or module) and 0 otherwise.

- a function F_k is a requirement that could be included in a finished product.
- a module M_j is an assembly of functions that could be added to 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 us introduce the following notations:

- $\mathcal{F} = \{F_1, \ldots, F_q\}$: the set of q functions that can appear in both finished products and modules;
- $\mathcal{P} = \{P_1, \ldots, P_n\}$: the set of n possible finished products that may be demanded by at least one customer. Note that D_i is the estimated demand of product P_i during the life cycle of the product family;
- $\mathcal{M} = \{M_1, \ldots, M_m\}$: the set of m possible modules.
- $\mathcal{S} = \{S_1, \ldots, S_s\}$: the set of s distant production facilities where a site S_l has a production capacity W_l.
- F_j^A : the fixed cost of module M_j at the nearby assembly facility (management costs);
- V_j^A : the variable cost of module M_j at the nearby assembly facility (cost of assembly, storage, transportation, etc.);
- F_{jl}^P : the fixed cost of module M_j at the distant production facility S_l (management)
- V_{jl}^P : the variable cost of module M_j at the distant production facility S_l (cost of assembly, storage, etc.);
- t_j : the time required to assemble module M_j in a finished product;
- T : the maximum assembly time available;
- W_{jl} : the work load generated by producing one module M_j at facility S_l.
- W_l : the work load capacity available at facility S_l.

Under these assumptions, a product (or module) is represented 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 $\mathcal{M}$ contains m modules. $\mathcal{M}$ may be all the possible modules in the whole combinatory, or a subset of those modules.

The problem is now to determine the subset $\mathcal{M}' \in \mathcal{M}$, of minimum cost, such that all products in $\mathcal{P}$ can be built in a constrained time window T. Concerning the products, the goal is to determine which bill of materials is the most suitable (Fig. 2).

In terms of the manufacturing process: (1) the producer assembly line costs must be minimized; and (2) the final assembly time must be less than the available time, in order to respect the delivery time for the customers. In terms of supply chain design: (1) each distant facility cost is considered (with fixed and variable costs for each possible module); and (2) the total workload at each production facility must be under its own production capacity.

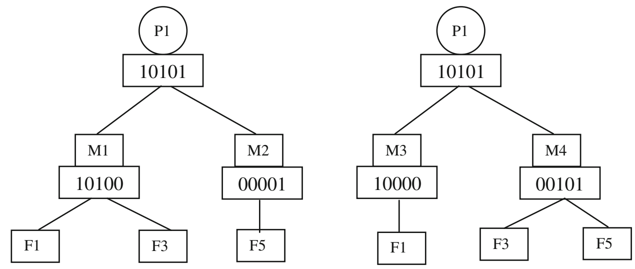


Fig. 2 Alternative bills of materials

The problem is modeled using a Mixed Integer Linear Program formulation. The objective is to minimize all the costs linked to the activities of the producer and suppliers. These costs are fixed, as a result of management of the modules at the nearby facility, assembly at the nearby facility, management of the modules at the distant facilities, and the production costs at the distant facilities.

Below, two strategies for solving the problem are proposed:

- A two-phase approach (2P_App) in which the design of modules precedes their assignment to production facilities,
- An integrated approach (In_App) in which all the costs are included in the mathematical model in order to obtain an optimal solution for both the selection of modules and their assignment to production facilities.

The following two sections present these strategies in greater detail.

A two-phase modeling approach

The basic idea in the first approach is to optimize costs separately. First, at the nearby facility, the bills of materials are drawn up, and the set of modules to produce is optimized. Second, the assignment of modules to the production facilities is optimized.

The first phase consists in determining the modules that optimize the assembly costs at the nearby facility, such that all finished products can be built within the constrained time window T:

$$Z^A = min\left(\sum_{j=1}^{m} F_j^A Y_j + \sum_{j=1}^{m} V_j^A \left(\sum_{i=1}^{n} D_i X_{ij}\right)\right) \quad (1)$$

$s.t.$

$$AX_i = P_i \quad \forall i \in \{1, \ldots, n\} \quad (2)$$

$$\sum_{j=1}^{m} t_j X_{ij} \leq T \quad \forall i \in \{1, \ldots, n\} \tag{3}$$

$$X_{ij} \leq Y_j \quad \forall i \in \{1, \ldots, n\}\ \forall j \in \{1, \ldots, m\} \tag{4}$$

$$Y_j, X_{ij} \in \{0, 1\} \quad \forall i \in \{1, \ldots, n\}\ \forall j \in \{1, \ldots, m\} \tag{5}$$

where A is the binary matrix, column j of which is the vector M_j; and X_i is the line vector composed by the variables X_{ij} and P_i a binary vector of size q; $X_{ij} = 1$, 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 (then Y_j belongs to $\mathcal{M}'$, the set of selected modules), 0 otherwise.

The objective function Z^A minimizes the costs incurred at the nearby facility, where $\left(\sum_{i=1}^{n} D_i X_{ij}\right)$ corresponds to the total demand of module M_j. Constraint (2) shows that a finished product P_i must be assembled exactly according to customer requirements. Constraint (3) indicates that products must be assembled within the time window T, in order to respect the delivery time. When T is long, constraint 3 is not active, and all the final products can be built directly from the elementary functions. Constraint (4) states that, if module M_j is used in the bill of materials of product P_i, then module M_j must be produced somewhere.

The problem described here contains the set-partitioning problem (Garey and Johnson 1979). We then conclude that it is NP-hard in the strong sense.

The second phase deals with the assignment of modules from the first phase on the distant facilities under capacity constraints:

$$Z^P = min\left(\sum_{l=1}^{s} \sum_{j|Y_j=1} F_{jl}^P Y_{jl} + \sum_{l=1}^{s} \sum_{j|Y_j=1} V_{jl}^P \left(\sum_{i=1}^{n} D_i X_{ij}\right) Z_{jl}\right) \tag{6}$$

$s.t.$

$$\sum_{l=1}^{s} Z_{jl} = 1 \quad \forall j | Y_j = 1 \tag{7}$$

$$\sum_{j|Y_j=1} W_{jl} \left(\sum_{i=1}^{n} D_i X_{ij}\right) Z_{jl} \leq W_l \quad \forall l \in \{1, \ldots, s\} \tag{8}$$

$$Z_{jl} \leq Y_{jl} \quad \forall j | Y_j = 1\ \forall l \in \{1, \ldots, s\} \tag{9}$$

$$Z_{jl} \geq 0 \quad \forall j | Y_j = 1\ \forall l \in \{1, \ldots, s\} \tag{10}$$

$$Y_{jl} \in \{0, 1\} \quad \forall j | Y_j = 1\ \forall l \in \{1, \ldots, s\} \tag{11}$$

where $Y_{jl} = 1$, if module M_j is produced at facility S_l, 0 otherwise; and Z_{jl} is the percentage of demand of module M_j produced at facility S_l.

The objective function Z^P minimizes the costs occurring at all distant location facilities. Constraint (7) indicates that the production of a module M_j must satisfy the overall quantities required. Constraint (8) shows that total production at facility S_l must not exceed that facility's capacity. Constraint (9) expresses the relation between the variables Z_{jl} and Y_{jl}. A module M_j can be produced at S_l only if M_j is assigned to $S_l (Y_{jl} = 1)$.

An integrated modeling approach

The second strategy consists in optimizing all costs at the same time, where the objective function Z_{opt} is the sum of the two-phase objective functions (Z^A and Z^P). Constraints are those of the two phases. In order to avoid the quadratic term $\left(\sum_{i=1}^{n} D_i X_{ij}\right) Z_{jl}$ in Z^P and in Eq. 8, we introduce the variable Q_{jl}, which represents the quantity of module M_j produced at site S_l and we suppress the variable Z_{jl}. Eq. 8 is replaced by:

$$\sum_{j=1}^{m} Q_{jl} \leq W_l \tag{12}$$

Equation 7 is replaced by:

$$\sum_{l=1}^{s} Q_{jl} = \sum_{l=1}^{s} D_i X_{ij} \tag{13}$$

Equation 14, in which B is a large number, replaces Eq. 9:

$$Q_{jl} \leq B Y_{jl} \tag{14}$$

The idea of such an approach is to make a global decision when designing both the products and the supply chain.

Computational experiments

Datasets, experimental conditions, and indicators

The objective of the experiments is to analyze the optimal solution behavior for several cost configurations and for different time windows T. For this, small instances have been randomly generated on which the set of possible modules, the finished product set, the distant facility set, the demands D_i, the assembly operating times t_j, and the distant production facility capacities are fixed, while the costs vary.

The experiments were conducted on a model with one assembly site (nearby location) and two production sites (distant locations) which compete for the production of the modules. The distant sites have limited production capacity, while the nearby site is assumed to have an unlimited production capacity, and so solutions are guaranteed to exist for any instance.

Assuming that the demand D_i of a product P_i is a decreasing function of the number of functions of the products, then, as soon as a finished product contains more options, the demand for it becomes lower than if it had fewer functions. The individual assembly operating times t_j are fixed to 1, so that constraint (2) results in a limitation in the number of modules for each bill of materials.

The fixed and variable costs associated with the bills of materials (F_j^A and V_j^A) are defined using a square-root function of q_j (the number of functions in module M_j). The assumption is that, in a mass customization context, when assembling all the various products, the complexity of the assembly operations depends on the number of functions; besides, adding another function to a product containing many functions is less expensive than doing so to a product with fewer functions.

The fixed and variable costs associated with production at the distant facilities are considered differently. The modules manufactured at the distant location could be produced for stock and shipped to the assembly plant (taking into account the information available about the consumption of each component). So, once the settings are finalized for a certain type of module, those settings could be saved for several modules. Production costs could be then be considered less dependent on the number of functions.

Then, the costs are defined as follows:

- $F_j^A = \alpha\,(\sqrt{q_j} + \lambda_1)$.
- $V_j^A = \beta\,(\sqrt{q_j} + \lambda_2)$.
- $F_{jl}^P = \gamma\, F_0^P$.
- $V_{jl}^P = \delta\, V_0^P$.

The coefficients α, β, γ and δ are used to scan different cost configurations. λ_1, λ_2 are jamming factors generated by a uniform probability law. F_0^P and V_0^P are also randomly generated.

Table 1 describes the parameter settings used to configure the various cost files for performing the tests and analysis. The columns show the settings of the twenty-seven cost files used in the tests. Each cost file is characterized by a specific ratio between the various problem costs. The first line shows the ratio between the first-phase (Assembly in the nearby plant) costs and the second-phase (Production in the distant facilities) costs. "A" indicates that the assembly costs are highly predominant, "C" indicates that the production costs predominate, while "B" indicates that assembly and production costs are almost equivalent. The second line shows the ratio between the fixed and variable costs of the assembly phase: "+" indicates that the fixed costs are higher, "−" indicates that the variable costs are higher, and "1" indicates that the costs are balanced. In the same way, the third line shows the relationship between the fixed and variable costs of the production phase. The remaining lines show the numerical values of α, β, γ and δ that correspond to each cost file. For example column (C1) shows that a problem described as (A, +, +) is one in which assembly costs predominate, i.e. high fixed costs in terms of both assembly and distant production. From a numerical point of view, the following parameters have been used ($\alpha = 1000$, $\beta = 0.10$, $\gamma = 1.44$ and $\delta = 0.01$).

For each of the 27 configurations, 10 instances have been generated. The problem data were fixed as follows: the number of functions $q = 8$, the number of finished products $n = 30$, where each product has at least $q_{min} = 3$ functions and at most $q_{max} = 6$ functions, $m = 255$ (all possible combinations of modules) and the number of production facilities $s = 2$. Jamming factors are generated with a uniform law, such that $8\% \leq \lambda_1, \lambda_2 \leq 12\%$; $50 \leq F_0^P \leq 100$; and $1 \leq V_0^P \leq 10$. T varies from 3 to 6. For $T > 6\,(q_{max})$ the solution is the same as for $T = 6$ i.e. the constraint 3 of the mathematical program is not active and the final products can be assembled without time constraint consideration. For $T \leq 2$, the final assembly will consider a maximum of 2 assembly operations for each final product, which does not seem reasonable from a practical point of view.

The tests were conducted in C++ with Ilog Cplex 9.0 library. They were solved on a 1.6 Hz DELL workstation with 512 Go of RAM.

In order to facilitate the results analysis, the following notations are introduced:

- CF^A represents the total fixed cost of assembly;
- CV^A represents the total variable cost of assembly;
- CF^P represents the total fixed cost of production;
- CV^P represents the total variable cost of production;
- $Z^A = CF^A + CV^A$ is the total cost of assembly;
- $Z^P = CF^P + CV^P$ is the total cost of production;
- $Z = Z^A + Z^P$ is the total cost for the two-phase approach (2P_App);
- Z_{opt} is the optimal total cost given by the integrated approach (Int_App);
- $Z_{opt}^A = CF_{opt}^A + CV_{opt}^A$ represents the total cost of assembly;
- $Z_{opt}^P = CF_{opt}^P + CV_{opt}^P$ represents the total production cost in the integrated approach.

Using these notations, the following indicators are used to analyze the experimental results obtained.

- $\Delta Z^A = \frac{Z^A - Z_{opt}^A}{Z_{opt}^A}$: the gap rate of Z^A between Int_App and 2P_App;
- $\Delta Z^P = \frac{Z^P - Z_{opt}^P}{Z_{opt}^P}$: the gap rate of Z^P between Int_App and 2P_App;
- $\Delta Z = \frac{Z - Z_{opt}}{Z_{opt}}$: the gap rate of Z between Int_App and 2P_App;

Table 1 Cost configurations

		A Costs								
		C1	C2	C3	C4	C5	C6	C7	C8	C9
Assembly costs % Production costs		A	A	A	A	A	A	A	A	A
Assembly costs	CF % CV	+	+	+	1	1	1	−	−	−
Production costs CF	% CV	+	1	−	+	1	−	+	1	−
Parameter's numerical values										
α		1000	600	200	240	120	120	100	60	20
β		0.10	0.10	0.10	0.40	0.15	0.20	0.50	0.30	0.10
γ		1.44	0.64	0.04	0.96	0.32	0.04	0.48	0.32	0.04
δ		0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
		B Costs								
		C10	C11	C12	C13	C14	C15	C16	C17	C18
Assembly costs % Production costs		B	B	B	B	B	B	B	B	B
Assembly costs	CF % CV	+	+	+	1	1	1	−	−	−
Production costs CF	% CV	+	1	−	+	1	−	+	1	−
Parameter's numerical values										
α		600	600	200	360	180	120	120	100	60
β		0.10	0.10	0.10	0.35	0.30	0.20	0.60	0.50	0.30
γ		9.60	4.80	0.40	7.20	3.20	0.40	4.80	3.20	0.40
δ		0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
		C Costs								
		C19	C20	C21	C22	C23	C24	C25	C26	C27
Assembly costs % Production costs		C	C	C	C	C	C	C	C	C
Assembly costs	CF % CV	+	+	+	1	1	1	−	−	−
Production costs CF	% CV	+	1	−	+	1	−	+	1	−
Parameter's numerical values										
α		400	400	200	120	240	120	20	20	20
β		0.20	0.40	0.10	0.20	0.40	0.30	0.10	0.10	0.10
γ		48	32	12	48	32	8	48	32	4
δ		0.25	0.5	3	1	1	2	1	1	1

- $|\mathcal{M}'|$: the number of the modules selected in $\mathcal{M}'$ (the solution size);
- Module requirement: the quantity of modules M_j required to assemble the finished products required: $Req_j = \sum_{i=1}^{n} D_i X_{ij}$
- Solution requirement: the sum of the requirements of the solution modules $\sum_{j=1}^{m} \sum_{i=1}^{n} D_i X_{ij}$

Analysis of the total cost

We first analyze how the total costs evolve according to cost structures and assembly time. Figure 3a–c show the gap between the results of 2P_App and Int_App. Since the integrated approach provides optimal results, the results are given such that 2P_App is represented as a percentage of Int_App. Figures 3a–c show the same diagrams with a reordering of the X-axis.

As we can see ΔZ^A is always negative because 2P_App gives an optimal solution for the assembly stage. Conversely, ΔZ^P and ΔZ are always positive because the solutions are better in terms of the whole supply chain, which is natural considering the optimization models.

Figure 3a allows us to see the tendency of the gaps when the second cost parameter moves from "−" to "1" to "+". We note that there is actually no clear tendency here, because the other cost parameters have a strong influence on the gap rate. However, from Fig. 3b, which shows the tendency of the gaps when the first cost parameter moves from "A" to "B" to "C", it is clear that ΔZ increases significantly when moving form "A" to "C" (Fig. 3c), since production costs take more importance in the global objective function.

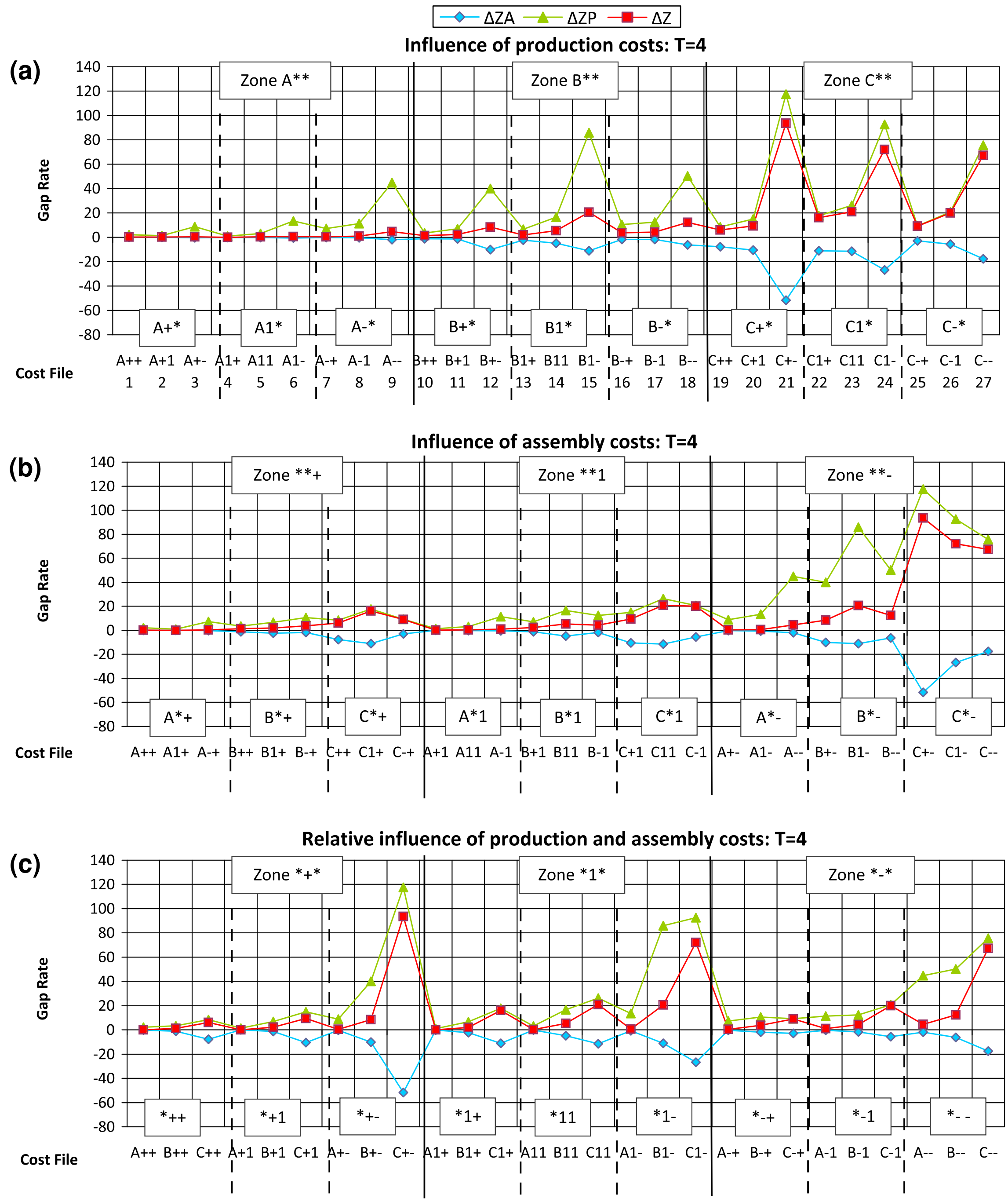


Fig. 3 Total cost (example with $T = 4$)

We also see that ΔZ shows a clear trend in the case of the third cost parameter. When the first two parameters are fixed, the gap rate ΔZ increases when the production cost parameter moves from "–" to "1" to "+". The amplitude of the gap increases progressively when moving from A to C, because the production costs take increasingly much more weight in the objective function, which allows Int_App to improve Z significantly compared with 2P_App.

Number of modules and total needs

In this section, we present the results for $T = 4$. The shape is similar for all other values of T. Figure 4 shows the solution

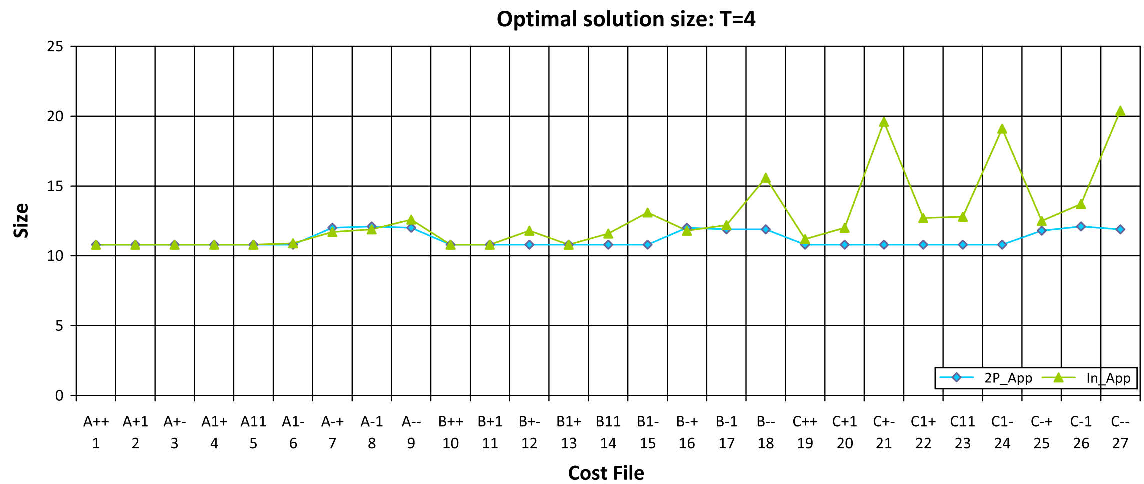


Fig. 4 Number of modules in $\mathcal{M}^{'}$ (example with $T = 4$)

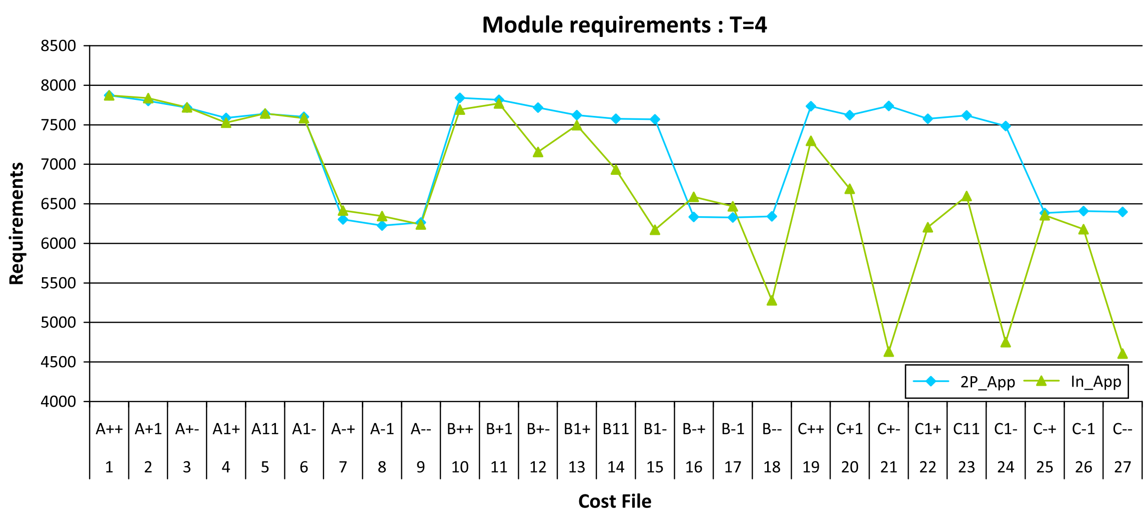


Fig. 5 Total needs (example with $T = 4$)

requirement, which is the number of modules in $\mathcal{M}'$ when $T = 4$.

Figure 5 shows the solution requirements when $T = 4$, also the same shape applies for various values of T. The solution requirement represents the sum of the needs of each module in $\mathcal{M}'$ to satisfy the demanded quantity of the family products.

For the same reasons, the solution size gaps and the solution requirement gaps follow the same trend according to the cost configurations, the only difference being that the solution sizes are bigger for Int_App than for 2P_App, while it is the opposite for the solution requirements.

Let us represent the cost configuration by *IJK*. Where $I \in \{A, B, C\}$ is the first cost parameter, $J \in \{+, 1, -\}$ is the assembly cost parameter, and $K \in \{+, 1, -\}$ is the production cost parameter.

For example, 2P_App always yields a small solution for cost files 1–6, 10–15 and 19–24; that is, when $J \neq$ "−" (i.e. when fixed assembly costs are greater than variable assembly costs). Obviously, this is to limit the number of modules used and so limit the fixed assembly costs, which represent the weight in the first-phase objective function.

If a solution contains a small number of modules, then each module has to satisfy more requirements (Fig. 5). We can see in this figure that the requirements corresponding to cost files 7, 8, 9, 16, 17, 18, 25, 26, 27 (for 2P_App) are lower than the requirements corresponding to the other cost files. This explains why the gap rate between 2P_App and Int_App is greater when $K =$ "−" (i.e. when variable production costs are greater than fixed production costs). In this case, 2P_App solutions have a relatively small number of modules (in order to minimize fixed assembly costs), and, consequently, the resulting modules will have more requirements, leading ultimately to a high value of CV^P after resolution of the second phase. In contrast, Int_App obtains a large solution directly, because it takes into account the variable production costs when determining the bills of materials.

Evolution of costs according to T

This section is aimed at analyzing the evolution of the different problem costs when T varies. For purposes of simplification, the results are presented for a single set of parameters. Moreover, the following conclusions can be generalized for the other configurations. Figure 6 shows this analysis for the problem using the C20 cost file ("C+1" structure, see Table 1). Figure 6 shows the results obtained with 2P_App in the first column, and the results with Int_App in the second column. The assembly, logistics, and total costs are detailed, as well as the solution size and solution requirements. This result holds for all the instances.

For both the assembly costs (phase 1) and the logistical costs (phase 2), the fixed costs decrease with T, while the variable costs increase, which leads, in most cases, to a reduction in Z^A and Z^P, and consequently in Z and Z_{opt}. The reason for this is that, when T increases, more time is available to assemble the final product from the modules, so it is possible to use a larger number of modules in each final product. Modules containing fewer functions are compatible with more final products, and a smaller number of modules could satisfy a larger number of products. An extreme case is where T is so large that it is possible to assemble all the final products from elementary functions. That is why we can observe that an increase in T leads to fewer size solutions (curves (d)), and consequently a decrease in total fixed costs (for both phases) (see curves (a) and (b)). In contrast, since the solution modules are used in the bill of materials of more products, their needs increase (curves (e)) and consequently the total variable costs increase. Generally, the total costs decrease with T for both approaches (curves (c)). However, sometimes it is not necessary to increase T to reduce the total costs, as we can see in curve 2 (c). This is because, for some cost configurations, the improvement in Z with an increase in T is negligible. This is valid for both approaches, since the configurations are such that the variable costs are much higher than the fixed costs, leading to a stagnation of Z at a certain point of T (Fig. 7). This result holds for all the instances.

Figure 8 shows a very important result, which is that, for some cost configurations, the total costs of 2P_App increase with T. Let us consider the case where production costs predominate over assembly costs, and the variable production costs are higher than the fixed production costs. Moreover, since the choice of modules to be used in the product family bills of materials has a major influence on the production costs, 2P_App could not succeed in reducing the total costs with an increasing T. At first, modules will be determined in the first phase so as to minimize Z^A, and, when the configuration is such that fixed assembly costs are greater than variable assembly costs, then increasing T leads to a reduction in the solution size and consequently to an increase in the solution requirements. This increase in requirements leads to an increase in the variable production costs following resolution of the second phase. For this reason, Z^P will certainly increase when T increases, and, since it represents a large proportion of the total costs, Z will increase with T. A representative shape is given for the instance $C2$ in Fig. 8.

Computational time

Examination of the computational time curves (Figs. 9 and 10) shows that the two-phase approach is extremely quick compared with the integrated approach. Generally, the two-phase approach is slower when the assembly cost parameter (J) is "−" (i.e. when variable assembly costs are greater than fixed assembly costs).

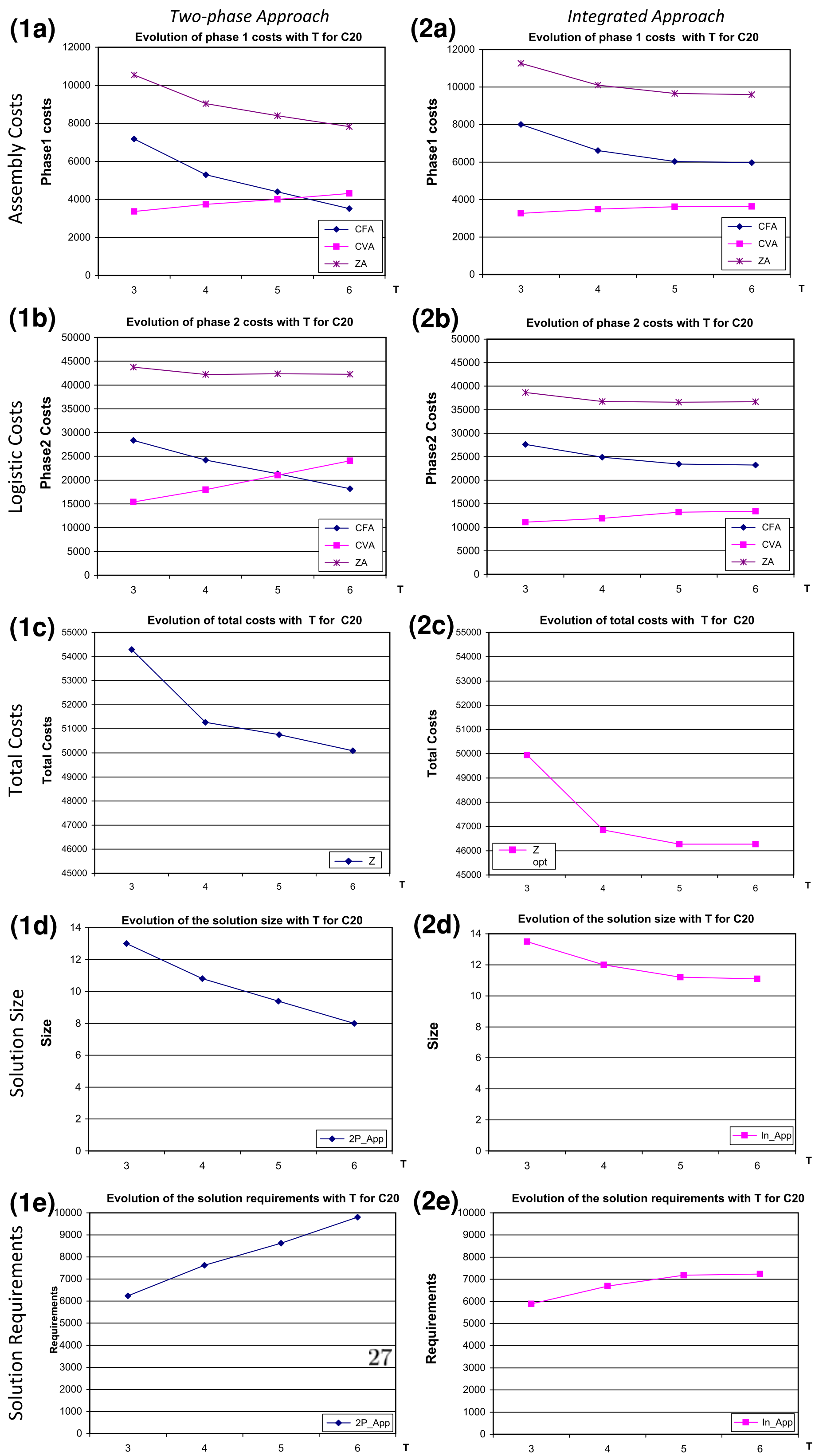
The integrated approach is much more time-consuming, especially when $K \neq$ "−" (i.e. when variable production costs are lower than fixed production costs). This phenomenon can be explained as follows: when $K =$ "−", CV^P is much higher than CF^P, then the solution must contain large modules to minimize requirements, and there is no special concern about the solution size, since CF^P is small. Hence, the MILP solver spots the interesting modules quickly (those having a low production costs) and builds the optimal solution. In contrast, when $K \neq$ "−", the solution size must not be so big that it minimizes CF^P, and this fact further complicates resolution of the problem, leading to a higher computational time. Uncharacteristically, when ($T = 3$), the computational time for costs is such that ($I =$ "B") is very high, because in this case constraint (2) is very difficult to tackle.

Finally, Fig. 10 shows the evolution of the computational time with T for the $C20$ configuration. Generally, the computational time decreases when T increases for both approaches, because the assembly time constraint becomes less difficult to respect. We also note in this figure the great reduction in computational time when T moves from 3 to 4.

Conclusion

This paper was dedicated to the difficult industrial problem that arises when companies attempt 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 of production, which is on the assembly line. Several authors have

Fig. 6 Evolution of costs with T (example with $C20$)



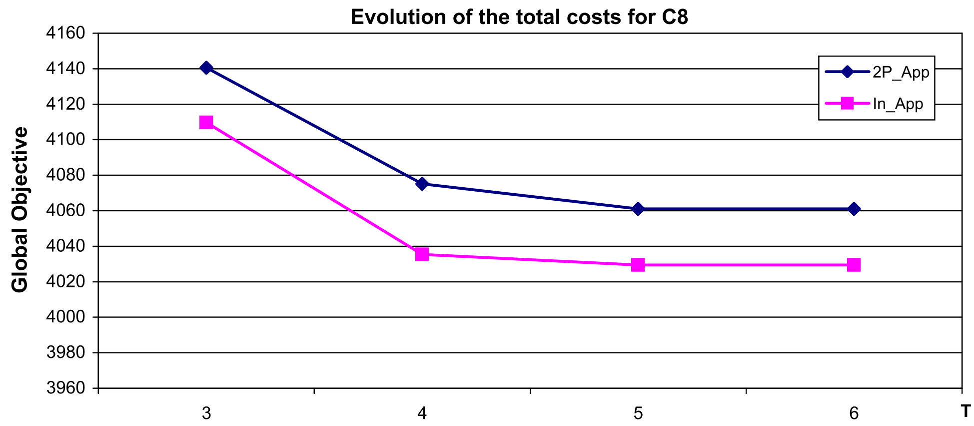


Fig. 7 Evolution of the total costs with T (example with $C8$)

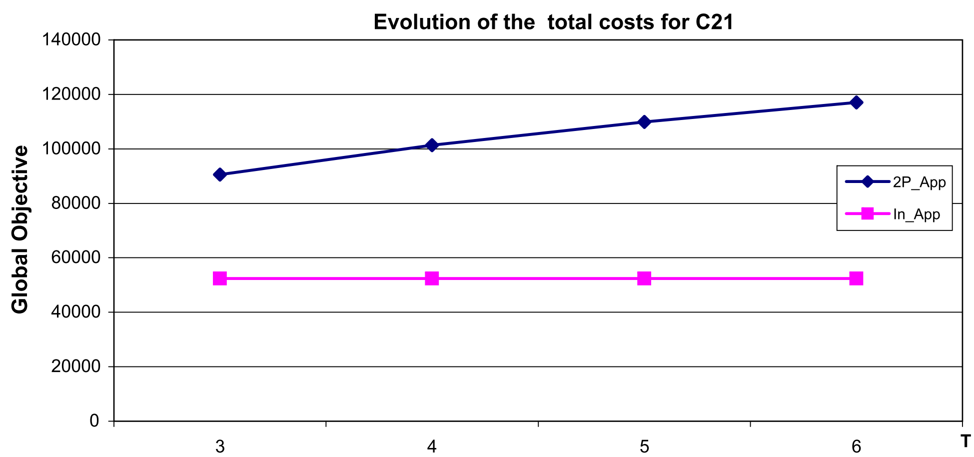


Fig. 8 Evolution of the total costs with T (example with $C21$)

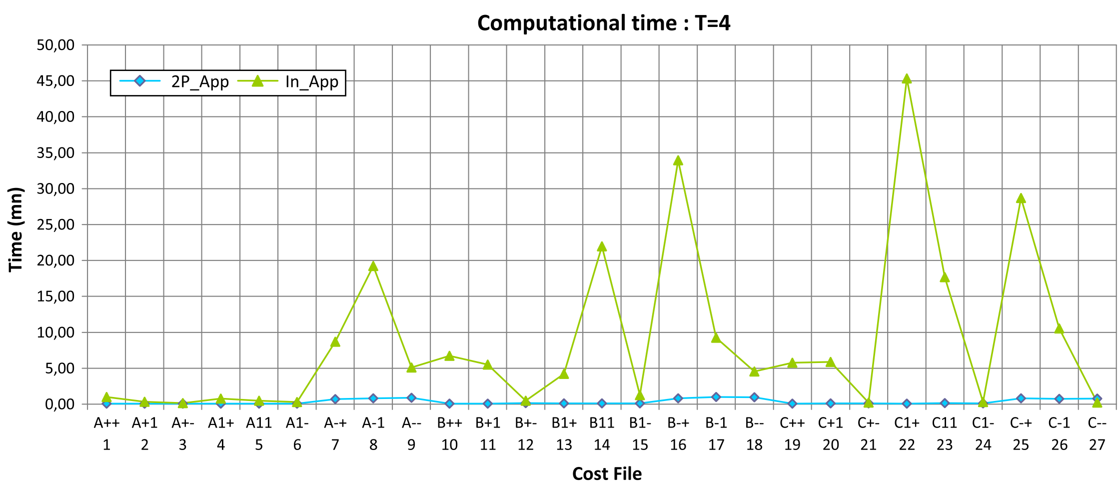


Fig. 9 Computational time (example with $T = 4$)

considered this problem based on 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 customer requirements.

We presented a new challenging model which simultaneously takes into account product family design, process design, and supply chain design. The product family design consideration is the determination of an efficient module set which allows products to be assembled while avoiding function redundancy. The process design consideration is constrained by delivery time requirements. Finally, the capacity constraints of distant facilities constitute the chief consideration for the supply chain.

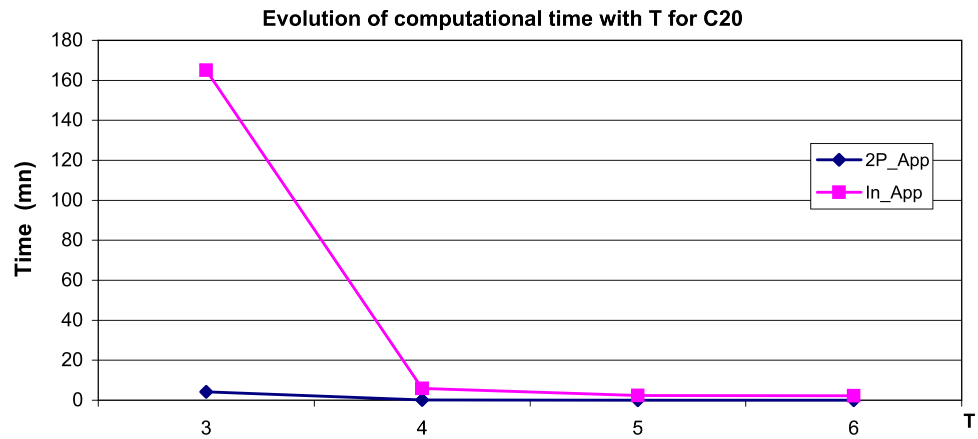


Fig. 10 Evolution of computational time with T (example with $C20$)

The model's objective functions are designed to optimize the costs incurred by the producer and the suppliers as a result of their activities. The main result is that the module architecture depends in particular on cost configurations between process and supply chain and also on delivery time.

Our tests confirm that the integrated approach is very much better than the two-phase approach when production costs predominate over assembly costs (C cost region), and when variable production costs are greater than fixed production costs. In contrast, in the A cost region, there is practically no gap between the results of the two approaches.

For the two-phase approach, the solution size increases when variable assembly costs are significant relative to the fixed assembly costs, while solution requirements decrease. This indicates that cost optimization favors small modules with big requirements (modules having a small number of functions which can be used in many products) when variable costs are low, and big modules with small requirements when variable costs are high. In contrast, the same phenomenon occurs for the integrated approach solution when production costs are significant relative to fixed production costs. The difference here is that the indicator of evolution is much more dramatic when the production costs at distant facilities exceed the assembly costs at the production facility close to the market. The two-phase approach is of interest for solving large problems, not only because it is better at doing so than the integrated approach, but also because it is much faster.

The analysis of the different problem indicators with the time constraint T for final assembly reveals that the two-phase approach tends to select small modules which can sometimes lead to a rise in the total costs when T increases. However, when variable costs are greater than fixed costs, increasing T has no effect on the total costs.

There are several future research areas to be explored with respect to this problem. It would be interesting, for example, to investigate the heuristics for larger cases where problem complexity becomes too great. It would also be interesting to study the influence of other parameters, like facility capacities and production strategies. Furthermore, we can consider the global model where there are many nearby facilities.

Many module assignment policies could also be analyzed:

- A module M_j could be produced at many distant facilities, which is the case of the model described above.
- The production of a module M_j is restricted in only one facility, in which case we have to add the following constraint: $\sum_{l=1}^{s} Y_{jl} = 1 \; \forall j | Y_j = 1$. This problem seems more difficult to solve due to the 0-1 assignment it contains.
- Every module must be produced at at least two facilities with a minimum percentage at each one. This is in order to anticipate production problems like delivery delay or worker strikes. Hence, we have to add the following two constraints: $Z_{jl} \geq \delta Y_{jl} \; \forall j | Y_j = 1 \; \forall l \in \{1, \ldots, s\}$ and $\sum_{l=1}^{s} Y_{jl} \geq 2 \; \forall j | Y_j = 1$. Again, this problem seems to be harder to solve.

References

Agard, B., & Penz, B. (2009). A simulated annealing method based on a clustering approach to determine bills of materials for a large product family. *International Journal of Production Economics, 117*(2), 389–401.

Agard, B., Cheung, B., & Da Cunha, C. (2006). Selection of a module stock composition using genetic algorithm. In *12th IFAC symposium on information control problems in manufacturing—INCOM 2006*, Saint-Etienne, France.

Amilhastre, J., Fargier, H., & Marquis, P. (2002).Consistency restoration and explanations in dynamic CSPs-application to configuration. *Artificial Intelligence, 135*(1–2), 199–234.

Bacuta, C. (2007). A unified approach for Uzawa algorithms. *SIAM-Journal-on-Numerical-Analysis, 45*(5), 2228–2258.

Briant, O., & Naddef, D. (2004). The optimal diversity management problem. *Operation Research, 52*(4), 515–526.

Da Cunha, C. (2004). *Définition et gestion de produits semi-finis en vue d'une production de type assemblage à la commande*. PhD thesis, INP Grenoble.

Da Cunha, C., Agard, B., & Kusiak, A. (2007). Design for cost : Module-based mass customization. *IEEE Transactions on Automation Science and Engineering, 4*(3), 350–359.

Daskin, M. S. (1995). *Network and discrete location, models, algorithms and applications*. New York: Wiley.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability, a guide to the theory of NP-completeness*. New York: W.H. Freeman and Company.

Hale, T. (2005). Trevor hale's location science references-available at the address. Technical report, http://www.ent.ohiou.edu/~thale/thlocation.html.

Huang, C. C., & Kusiak, A. (1998). Modularity in design of products and systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 28*(1), 66–77.

Huei-Chuen, H., & Rongheng, L. (2008). A k-product uncapacitated facility location problem. *European Journal of Operational Research, 185*, 552–562.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical-Review, 106*(4), 620–630.

Jaynes, E. T. (1982). On the rationale of maximum-entropy methods. *Proceedings-of-the-IEEE, 70*(9), 939–9524.

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*(1), 1030–1047.

Pine II, B. J. (1993). *Mass customization: The new frontier in business competition*. Boston: Harvard Business School Press.

Xu, X., Fang, S. L., & Gu, X. J. (2007). Mass customization and variety. *Computer Integrated Manufacturing systems, 13*(7), 1330–1335.

Yang, Y., Wang, J. X., & Sang, S. J. (2007). Research on product family design for mass customization. *Modular Machine Tool and Automatic Manufacturing Technique*, (10), 9–13.

Zhiuong, W., & Fanqhua, N. (2007). Study on supply chain management for product family in mass customization. In *IEEE international conference on automation and logistics* (pp. 1515–1519).