DESIGN FOR COST: FAST HEURISTICS TO ASSIGN MODULES TO MANUFACTURING FACILITIES IN A PRODUCT FAMILY

Bruno Agard[1], Bernard Penz[2], Frederic Schairer[1, 2]
[1]*CIRRELT, Département de Mathématiques et de Génie Industriel,*
École Polytechnique de Montréal
C.P. 6079, succ. Centre-ville, Montréal (Québec), H3C3A7, Canada
bruno.agard@polymtl.ca
[2]*G-SCOP, Grenoble INP – CNRS - UJF*
46 avenue Félix Viallet, 38031 Grenoble Cedex 1, France
bernard.penz@g-scop.inpg.fr

Abstract: In a business world influenced by globalisation and technological progress, products are highly customized to suit each customer's demand. Companies propose a large diversity of products in order to satisfy existing customers' needs and to attract new customers. The strategy implemented by the industry to support such diversity involves creating product families. In a product family, subsets of components (called modules) are designed whereby the inventory costs can be reduced and the delivery time met. This paper presents a general problem about the simultaneous design of a product family and its supply chain. For this, a binary linear programming model is investigated. This contribution is about the assignment of modules to manufacturing facilities. The main objective is to design fast heuristics which optimize the costs of the resulting supply chain. To solve the assignment problem, heuristics have been developed and tested. To demonstrate the performance of these heuristics, various tests have been carried out. Computational results proved that the proposed heuristics, if combined are efficient to solve large size problems. This is a part of a more general problem that considers the selection of the set of modules.

Keywords: supply chain design, product family design, assignment problem, heuristics.

1 Introduction

In a business world influenced by globalisation and technological progress, companies are exposed to high competition and are forced to offer a large diversity of products in order to satisfy existing customers' needs and to attract new customers. As a result, consumers are becoming more demanding as they are spoilt for choice by the increasing range of products available to them [10].

For the companies, this mass customization creates a huge diversity of products and product components that is difficult to manage. On the one hand, ordered products have to be manufactured and delivered on time. On the other hand, inventory costs have to be reduced to stay competitive. Standardisation is a strategy largely employed [3]. A different strategy implemented by the industry involves creating product families [11]. In a product family, subsets of components (called modules) are designed whereby the inventory costs can be reduced and the delivery time met [12].

In this context, this paper presents a general problem about the simultaneous design of a product family and its supply chain. For this, a binary linear programming model is investigated. The contribution of this paper is about the assignment of modules to manufacturing facilities. The assignment problem is a part of a more general problem that considers the selection of the set of modules. The whole problem is presented, the selection of a relevant set of modules is not investigated here, previous results are extended with the assignment of those modules to manufacturing facilities. Since the search for modules is based on meta-heuristics, the focus is on developing FAST heuristics. That heuristics will be linked in the optimization process for the selection of modules.

Section 2 describes the state of the art. Section 3 presents the complete mathematical model and notations. The mathematical model deals with the selection of subassemblies (modules) for the product family and with the assignment of those modules to production facilities. Section 4 presents options to solve the assignment problem, exact solutions are available for small problems and since the problem is NP-Hard, two heuristics are proposed for larger problems. Section 5 gives the results for a set of experiments. Finally section 6 concludes the paper and proposes further investigations.

2 State of the art

The design of product families received a lot of attention in both the literature and practical specialists [11]. The design of a product family consists at satisfying a wide diversity of customers' requirements with a limited and rationalized product structure [4].

A product family is most often supported with a product platform that permit to share common elements (parts, subassemblies, processes…) and take advantage of economies of scales [4]. This is advantageously combined with postponement which consist of delaying as far as possible the moment when to products have to be considered different on the manufacturing process [9]. Each product from the product platform can be customized to satisfy each individual requirement. Modular design permits to produce large diversity of products with a limited number of standard elements, called modules [6, 7].

During the last years, firms developed a strategy that consists of producing modules in low cost countries, and assembling the final products close to the customer market [5]. This strategy leads to think differently the design of the product family, and to look for suppliers able to produce the modules. In this context, the production and transportation costs are both important.

Few academic results are available that consider simultaneously the design of the product family and the supply chain. Nevertheless, Lamothe et al. [8] proposed an optimization model for cost optimization by selecting a product family and designing its supply chain, based on a generic bill of material. One of the objective is also to reduce (or bound) the assembly time in the assembly factory [1].

The product family we try to design is composed by a great number of finished products, differing by the functions they contain. A finished product is view as a collection of functions. A module is then a collection of elementary functions appearing in at least one final product.

3 Mathematical model

The product family we try to design is composed by a great number of finished products, differing by the functions they contain. A finished product is view as a collection of

functions. A module is then a collection of elementary functions appearing in at least one finished product.
The problem is to select the best set of modules that will permit to manufacture every final product; simultaneously the problem is to define where the modules will be produced. We consider in this study that a module is produced in only one site, in order to reduce the fixed costs associated to the management of a module in a site. The number of possible modules is exponential in the number of functions, and then intractable in a decision aid tool. We propose a mathematical model that supports simultaneously the design of a bill of materials for all the products and the assignment of the modules to different production facilities.
After introducing some notations and parameters, we define the decision variables and propose a Binary Integer Linear Programming formulation.

3.1 Notations

Following notations are considered in the mathematical modeling:
Indices:

- i: index of functions $(1 \le i \le F)$, where F is the number of all the functions possibly assembled in a finished product.
- j: index of modules $(1 \le j \le M)$, where M is the size of the predefined set of modules (potentially exponential in the number of functions).
- k: index of finished products $(1 \le k \le P)$, where P is the number of finished product (given by the marketing department).
- l: index of production sites $(1 \le l \le S)$, where S is the number of all possible suppliers in the supply chain, and site 0 is the final assembly site.

Parameters

- tm_{jk} : processing time to assemble module j in product k.
- tf_{ik} : processing time to assemble function i in product k, when function i is not contained in a pre-assembled module.
- CM_{jl} : Fixed cost to manage module j in site l. This cost includes recurrent production costs, investment to adapt the production system, investment to prepare equipment for the transportation to the assembly factory and other costs. This evaluated cost concern the launching of the module production and not the day to day costs (labor and material costs depending on the number of items to produce), it is why it is independent of the number of produced module over time.
- CF_{il} : Fixed cost to pay each time the function i appeared in a module j produced in site l. This cost includes management cost of function i, inventory management fixed costs, procurement costs, etc.
- WF_{il} : Workload to manage function i in site l. This workload is paid each time the function i occurred in a module j produced at site l.
- CW_l : Workload capacity of site l. This capacity is defined to take into account the maximum quantity of function-modules a site is able to manage simultaneously. This is independent of the quantity of products (modules and functions) the site has to process day by day.

- $\delta_{ij} = 1$ if the function *i* is in module *j (*given information).
- $\gamma_{ik} = 1$ if function *i* is in product *k*, 0 otherwise (given information).
- $\lambda_{jk} = 1$ if the module *j* can be included in product *k (*given information).
- T : Maximum assembly time at site 0

Decision variables

- $x_{jk} = 1$ if module *j* is included in the bill of material of product *k*, 0 otherwise.
- $y_{jl} = 1$ if module *j* is produced in site *l*, 0 otherwise.
- $z_{ik} = 1$ if function *i* does not appear in a module, and is directly assembled during the final assembly process in product *k*.

3.2 Mathematical modeling

The described problem can be formulated as follows:

$$\min \left(\sum_{l=1}^{S} \sum_{j=1}^{M} CM_{jl} y_{jl} + \sum_{l=1}^{S} \sum_{i=1}^{F} \left(CF_{il} \sum_{j \mid \delta_{ij}=1} y_{jl} \right) \right)$$

s.t.

$$z_{ik} + \sum_{j \mid \delta_{ij}=1} x_{jk} = \gamma_{ik} \; ; \quad forall\ k \in \{1,...,P\} \tag{1}$$

$$x_{jk} \leq \sum_{l=1}^{S} y_{jl} \; ; \; forall\ \ j \in \{1,...,M\}\ and\ k \,\middle|\, \lambda_{jk} = 1 \tag{2}$$

$$\sum_{l=1}^{S} y_{jl} \leq 1 \; ; \; forall\ \ j \in \{1,...,M\} \tag{3}$$

$$\sum_{i=1}^{F} WF_{il} \left(\sum_{j \mid \delta_{ij}=1} y_{il} \right) \leq CW_l \; ; \; forall\ l \in \{1,...,S\} \tag{4}$$

$$\sum_{j \mid \lambda_{jk}=1} tm_{jk} \times x_{jk} + \sum_{i \mid \gamma_{ik}=1} tf_{ik} \times z_{ik} \leq T \; ; \; forall\ k \in \{1,...P\} \tag{5}$$

$$x_{jk}, y_{jl,} z_{ik} \in \{0,1\}; \forall j, \forall k, \forall l \tag{6}$$

Constraint (1) guarantees that for each function *i* of each product *k*, the bill of material contains exactly one time the function *i*, in a module *j* or is directly assembled at the final assembly process. Constraint (2) guarantees that a module used in at least one finished product must be built in a production site and constraint (3) guarantees that a module is produced in only one site. Constraint (4) verifies that the capacity of each site is verified and constraint (5) that the final assembly time is less or equal than the predefined bound. Finally constraints (6) impose that variables x_{jk}, y_{jl} and z_{ik} are binary.

We remark that variables z_{ik} can be removed from the model because it is possible to deduce them when variables x_{jk} and y_{jl} are fixed. The problem is then to determine the value of each variable x_{jk} and y_{jl}. x_{jk} considers the selection of a set of modules for the product family (it correspond to design a bill of material for the product family) and y_{jl} correspond to the assignment problem.

The selection of modules (x_{jk}) is NP-hard and some meta-heuristics have already been proposed [2, 5]). In this paper we propose some heuristics for the assignment problem (y_{jl}).

4 Solving the assignment problem

The assignment problem consists of the selection of a production facility for each module that is selected for the design of the product family. Each module has to be assigned in only one facility and each facility has a total workload limit. In the following, we suppose that x_{jk} decision variables are fixed, the bill of materials (or the set of modules selected) for the product family is known. The assignment problem consists of selecting the y_{il} variables. Three solution options are proposed: the first one is based on a branch-and-bound and gives optimal solutions. However its computing time is long and not suitable for large number of modules. Furthermore, this assignment problem must be solved frequently when designers search good bills of material for the product family. Consequently, two heuristics are then proposed and evaluated.

For the assignment problem the modeling may be adapted as follows: fixed costs CM_{jl} and CF_{il} are merged in $CT_{jl,}$, then

$$\sum_{l=1}^{S}\sum_{j=1}^{M} CM_{jl}\, y_{jl} + \sum_{l=1}^{S}\sum_{i=1}^{F} \delta_{ij}\left(CF_{il} \sum_{j \mid \delta_{ij}=1} y_{jl} \right) = \sum_{l=1}^{S}\sum_{j=1}^{M} CT_{jl}\, y_{jl}$$

with $CT_{jl} = CM_{jl} + \sum_{i=1}^{F} \delta_{ij} CF_{il}$

The problem becomes:

$$\min \quad \sum_{l=1}^{S}\sum_{j=1}^{M} CT_{jl}\, y_{jl}$$

s.t.

$$\sum_{l=1}^{S} y_{jl} = x_j \qquad \textit{forall } j \in \{1,\dots,M\} \qquad (7)$$

$$\sum_{i=1}^{F} WF_{il}\left(\sum_{j \mid \delta_{ij}=1} y_{jl} \right) \le CW_l \qquad \textit{forall } l \in \{1,\dots S\} \qquad (8)$$

$$y_{jl} \in \{0,1\} \qquad \forall j, \forall l \qquad (9)$$

Constraint (7) implies that each module is assigned to a production facility where x_j=1 if module j is selected in a bill of material, 0 otherwise. Constraint (8) permits to respect each production facility workload, and constraint (9) insures that each module is assigned in only one production facility. Without constraint (8), the problem is trivial because an optimal solution assigns each module to the cheapest site. Unfortunately constraint (8) makes the problem NP-hard. An obvious reduction from the well known Bin Packing Problem proves this complexity result

4.1 Branch-and-bound procedure

For small instances, it is possible to evaluate an optimal solution with a MILP solver. The goal of the paper is to evaluate fast heuristics for large instances, in order to use them in more complex optimization models. It is why we choose to use the standard *bintprog()* procedure from the *Matlab* software for reference.

4.2 Greedy heuristics

Two heuristics are proposed for the assignment problem. Both are based on matrix CT_{jl} (see Figure 1). The first one (cost-module-facility) considers modules one by one and try to assign them to the best site. The second one (cost-facility-module) considers sites one by one and try to assign best modules in them.

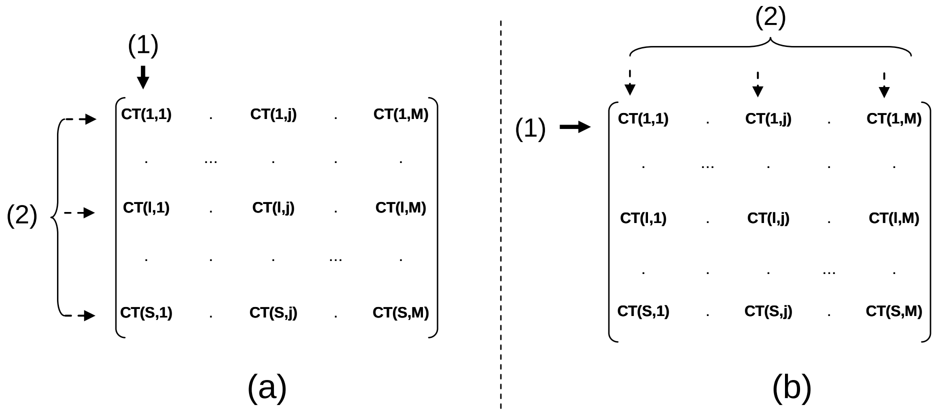

Figure 1: Heuristic cost-module-facility (a) and heuristic cost-facility-module (b)

4.2.1 Cost-module-facility heuristic

This heuristic selects for each module, one after the other, the production facility with the minimum cost (CT_{jl}) until the facility is full, otherwise it selects the second best cost, and so on (Figure 1, (a)). The algorithm is the following (With: $WF_{jl} = \sum_{i=1}^{F} \delta_{ij} \times WF_{il}$):

```
Algorithm Heuristic cost-module-facility
j: index for module
l: index for site
Total_assignment_cost=0
For all l, W[l]=0
For each module j
    While some site are not full
        Select site l with minimum cost for j
        Compute WF[j,l]
        If W[l] + WF[j,l] <= CW[l]
             Y[j,l] ← 1 (Module j is assigned to site l)
             W[l] ← W[l] + WF[j,l]
             Total_assignment_cost ← Total_assignment_cost + CT[j,l]
             break
        Else
             Site l is considered full for j
        EndIf
    EndWhile
    If j can not be affected to any facility
        Total_assignment_cost ← Total_assignment_cost + Penalty
    EndIf
EndFor
Return Total_assignment_cost
EndAlgorithm
```

4.2.2 Cost-facility-module heuristic

The cost-facility-module heuristic is similar to the previous one. Besides it considers each site, one by one, to assign all possible modules instead of considering each module for assignment (Figure 1, (b)). The algorithm follows:

```
Algorithm Cost-facility-module
j: index for module
l: index for site
Total_assignment_cost=0
For all l, W[l]=0
For all j,l Y[j,l]=0
While each site l is not full and some modules need assignment
    Select module j not yet assigned with minimum cost for l
    If W[l] + WF[j,l] <= CW[l]
       Y[j,l] ← 1 (Module j is assigned to site l)
       W[l] ← W[l] + WF[j,l]
       Total_assignment_cost ← Total_assignment_cost + CT[j,l]
       break
    Else
       Y[j,l] ← -1 (Module j is not possible for site l)
    EndIf
EndWhile
For j=1 to M
    If can not be affected to any facility
        CT[j,l] ← CT[j,l] + Penalty
    EndIf
EndFor
EndAlgorithm
```

5 Experiments

Experiments are realized on a Pentium 4, 3GHz with 512Mo RAM. The programming environment is *Matlab*. The Branch and Bound used to obtain an optimal solution is the standard *Bintprog()* procedure.

The problem considers the assignment of modules to production facilities, each module contains up to 13 different functions. Three (3) manufacturing facilities are considered. One of the production facilities is for final assembly. In all experiments production capacity of the final assembly site is considered to be unlimited (in practice CW_0 = 10 000). In the following, production capacities of the manufacturing facilities vary, as well as costs and the number of modules to assign.

The overall design problem (selection of modules and assignment) is decomposed in two steps. The first step permits to select a set of modules that are necessary for the design of the product family, without assignment. Previous research results permit to obtain such result ([5], [2]). In the second step the modules selected previously are assigned to the production facilities. The focus of this paper is on the validation of the assignment procedure. For each numerical test, the same set of modules is considered for all assignment procedures.

5.1 Variation on the number of modules

This experiments show the comportment of both heuristics compared to optimal results, obtained from the branch and bound procedure for solving various size problems.

Computational results show that for small size problems all methods are able to provide answers in a short time. Besides when the number of modules to assign increases (around 100 modules), computational time for exact solution explodes, and may not give any answer. Also it appears that production capacity of the different facilities have an influence on computational time. When production capacities are limited, computational time for exact solution also explodes. Computational time for both heuristics do almost not vary with the number of modules to assign and is always quasi instantaneous.

In the following, two examples are considered. In the first one, Ex. 1 (on Figure 2), production capacities of both distant sites are equal to 1000 (which is relatively high for generated instances); in the second example, Ex. 2 (on Figure 3), production capacity of facility 1 is reduced (500) and becomes a constraint for the assignment problem.

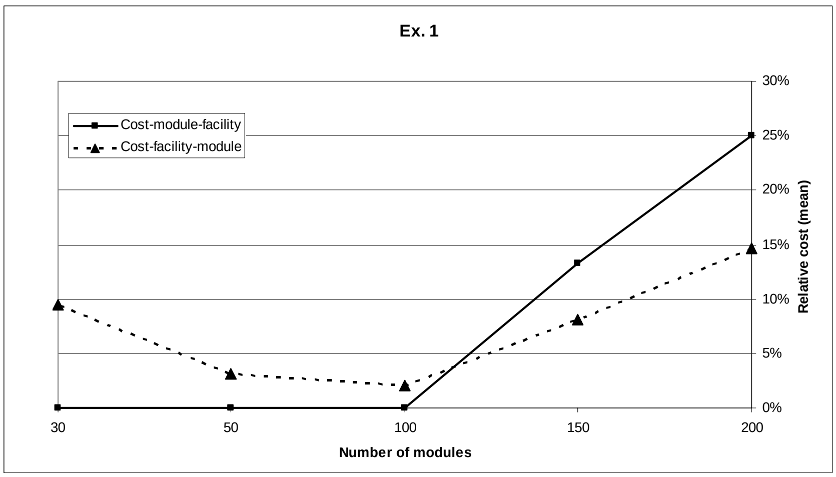

Figure 2: Assignment cost for Ex 1.

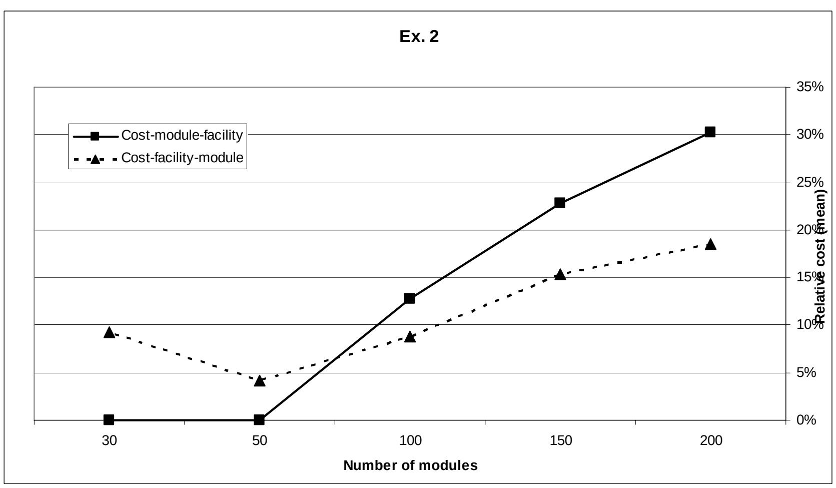

Figure 3: Assignment cost for Ex 2.

Obviously proposed heuristics are less performing than the exact method. On small size problems (100 modules and less), the difference with the best heuristic is less than 10 percent of the optimal cost. Besides, the best heuristic goes up to 20% for large size problems. Nevertheless both heuristics give an answer quasi instantaneous.

5.2 Variation on production capacities

Different experiments are conducted to evaluate the behaviour of the heuristics according to various production capacities.

Different tests are considered, total production capacity for the supply chain is fixed for each set of tests but production capacity varies from site to site. In all cases, production capacity for the final assembly facility is fixed (10 000). Figure 4 considers a total production capacity of 11 000 (large production capacity), 10 500 for Figure 5 (medium production capacity) and 10 200 for Figure 6 (limited production capacity). Two production facilities are available for the assignment of modules. Notations on x-axis express production capacity for site 1, production capacity for site 2, and production capacity for final assembly facility. The number of modules to assign is fixed (18 modules). Results from the branch and bound method are the reference, relative performance of each heuristics is provided.

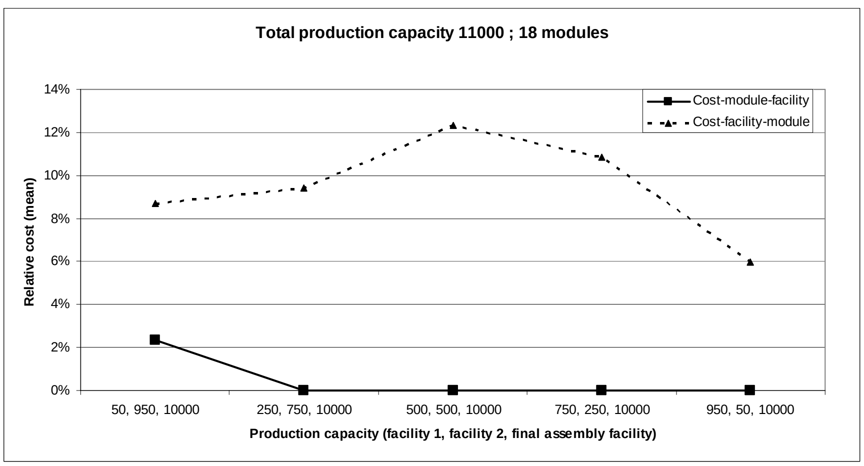

Figure 4: Assignment cost for large production capacity.

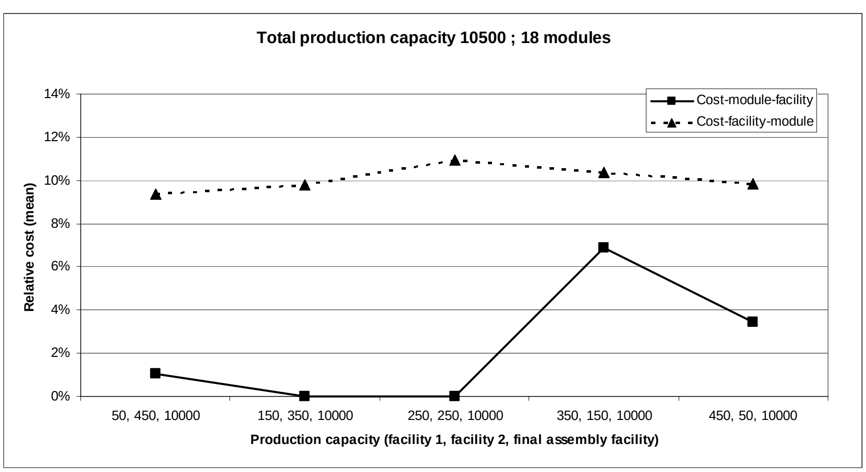

Figure 5: Assignment cost for medium production capacity.

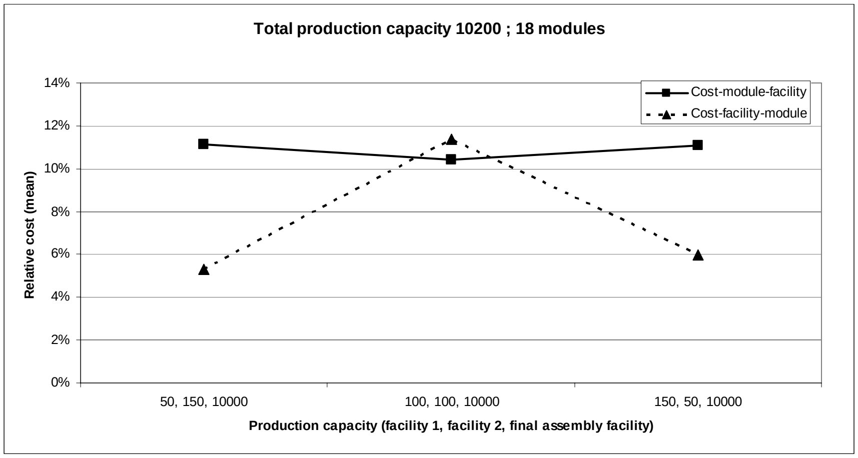

Figure 6: Assignment cost for limited production capacity.

These experiments prove that the cost-module-facility heuristic gives quasi-optimal results for the assignment of modules when production capacities are large. Besides when total production capacity decreases, less performing results are observed. Also repartition of production capacity has an influence. The cost-facility-module heuristic seems to be less performing, nevertheless its performance is almost stable when total production capacity varies. When production capacities are limited (especially if the production capacity of one site is limited) the cost-facility-modules becomes more performing; this heuristic takes into account the production workload as a first criteria.

5.3 Variation on production costs

In these experiments different production costs are tested for large and limited production capacities. Notation 1, 2, 1, for example, means that production costs in site 1 is equal to 1 $unit for any kind of module, while the production costs are equal to 2 $unit for site 2 and 1 $unit for the final assembly facility. 18 modules have to be assigned for each problem.

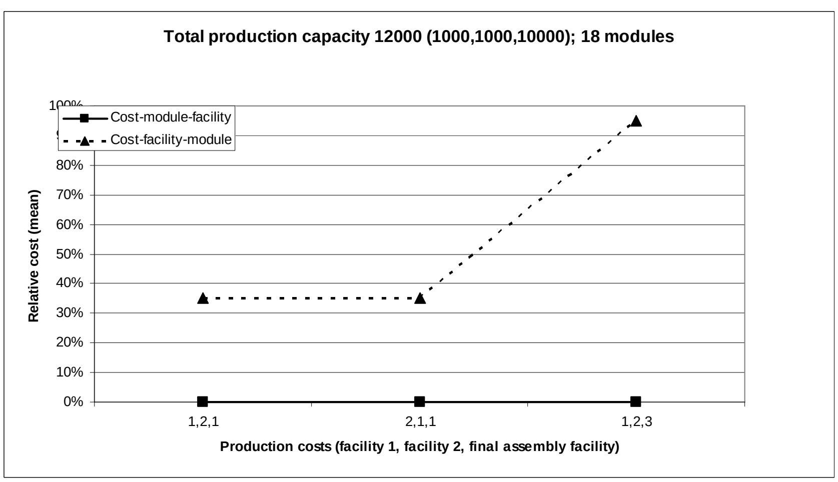

Figure 7: Assignment cost for large production capacity and various production costs.

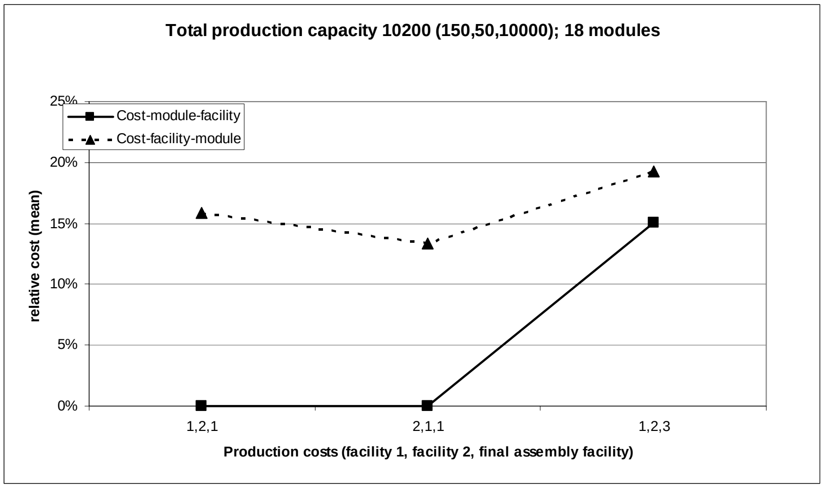

Figure 8: Assignment cost for limited production capacity and various production costs.

This shows that for large production capacities, the cost-module-facility always gives optimal results (the number of modules is equal to 18). For limited capacities we can observe that if the final assembly facility costs are high this heuristic loses performance; that comes form the greedy approach, some modules are not able to be assigned in others sites and the final assembly site has to manufacture them.

6 Conclusion and future work

Previous experiments proved that an optimal solution is not possible for computational time reason when solving large size instances and that heuristics may be advantageously used. Two heuristics have been proposed and evaluated. Both heuristics are quasi instantaneous for assignment of various size problems but may give less performing assignment costs.

For the general problem, exposed in section 3.2, it is not possible to solve it to the optimal (NP-hard). Presently some meta-heuristics are available to solve the selection of modules (x_{ij} variables). It is then possible to solve the general problem in two ways. On the one hand, the problem is separated, x_{ij} variables are computed, and then assignment is done once; in that case it is suggested to use an exact solution for the assignment problem. On the other hand the selection and assignment of modules are integrated, it means that for each iteration of the meta-heuristic approach a fast evaluation of assignment is needed; in that case both heuristics should be used and the best answer selected; after the last integrated iteration it is possible to improve the solution with the use of the exact solution once.

Future work will consider the solving of the general problem, focusing of the design of the product family and the supply chain simultaneously.

7 References

[1] B. Agard, A. Kusiak, Data Mining for subassembly selection, *ASME Transactions: Journal of Manufacturing Science and Engineering*, Vol. 126, No. 3, pp. 627-631, 2004.

[2] B. Agard, B. Cheung, C. da Cunha, Selection of a modules stock composition using genetic algorithm, *12th IFAC Symposium on Information Control Problems in Manufacturing – INCOM 2006*, Saint-Etienne, France, May 17 – 19, 2006.

[3] O. Briant and D. Naddef. The optimal diversity management problem, *Operations Research*, Vol. 52, No. 4, pp. 515-526, 2004.

[4] P. Child, R. Diederichs, F.H. Sanders, and S. Wisniowski, The management of complexity, *Sloan Manag. Rev.*, Vol. 33, No. 1, pp. 73–80, Fall, 1991.

[5] C. da Cunha, B. Agard, A. Kusiak, Design for cost: Module-Based Mass Customization, *IEEE Transactions on Automation Science and Engineering*, Vol. 4, No. 3, pp. 350-359, 2007.

[6] C.C. Huang and A. Kusiak, Modularity in design of products and systems, *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, Vol. 28, No. 1, pp. 66–77, 1998.

[7] A. Kusiak, *Engineering Design: Products, Processes, and Systems*, Academic Press, San Diego, CA, 1999.

[8] J. Lamothe, K. Hadj-Hamou, and M. Aldanondo. An optimization model for selecting a product family and designing its supply chain, *European Journal of Operational Research*, Vol. 169, No. 1, pp. 1030-1047, 2006.

[9] H.L. Lee and C.S. Tang, Modeling the costs and benefits of delayed product differentiation, *Management Science*, Vol. 43, No. 1, pp. 40–53, 1997.

[10] B.J. Pine II. Mass Customization: The New Frontier in Business Competition, *Harvard Business School Press*, Boston, 1993.

[11] T.W. Simpson, Z. Siddique, and R.J. Jiao (Eds), *Product Platform and Product Family Design - Methods and Applications*, Springer, 2006.

[12] J.P. Womack, D.T. Jones, and D.Roos, *The Machine That Changed the World*, Rawson Associates, 1990.