# CONSTITUTION D'UN STOCK DE PRODUITS SEMI FINIS POUR UNE DEMANDE DE PRODUITS VARIÉS SOUS CONTRAINTE DE TEMPS D'ASSEMBLAGE FINAL AVEC RECUIT SIMULÉ

**Catherine da Cunha, Bruno Agard**
Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal,
C.P. 6079, succ. Centre-ville, Montréal (Québec), H3C3A7, Canada
catherine.dacunha@polymtl.ca, bruno.agard@polymtl.ca

**Résumé :**

*Afin de proposer une grande diversité de produits finis à moindre coût, de nombreux industriels utilisent la conception modulaire. Ceci permet de développer et fabriquer indépendamment les différents modules qui seront ensuite assemblés à la commande. La conception modulaire des produits a des implications importantes tout au long de la chaîne logistique, des fournisseurs jusqu'aux clients. Lors du choix des modules à stocker, il est donc nécessaire de considérer les coûts induits par cette conception et ce aux différentes étapes de la production. La recherche d'une définition optimale des modules doit néanmoins se faire sous contrainte d'un temps de mise à disposition du produit fini qui soit acceptable pour le client.*

*Cet article propose l'utilisation d'un recuit simulé pour déterminer la composition d'un stock de modules. Le stock de modules doit permettre l'assemblage de n'importe quel produit fini sans aucune standardisation. De plus, la composition du stock de modules devra minimiser les coûts de la chaîne logistique (fabrication, transport et stockage) en respectant une contrainte de temps d'assemblage final. Les résultats obtenus par simulation montrent que cette démarche est pertinente. Les stocks de produits semi-finis obtenus permettent de respecter la contrainte de temps d'assemblage final et les coûts globaux sont optimisés.*

**Mots clés: conception modulaire, assemblage à la commande, famille de produits, recuit simulé.**

## 1 Introduction

Un marché fortement concurrentiel entraîne de la part des clients des exigences de qualité plus élevées, des exigences de services améliorés ainsi qu'une considération spécifique de leurs besoins. Les clients veulent recevoir le produit répondant exactement à leur besoin, avec un service offert de qualité, au coût minimum et cela immédiatement [1].

Pour faire face à cela, les industriels doivent mettre sur le marché des produits au plus près des besoins de chaque client et ceci avec un temps de mise à disposition minimum. La grande diversité des produits finis et le faible temps de mise à disposition des produits personnalisés entraînent une situation inconfortable. Pour faire face à la diversité des produits, l'industriel peut être tenté de faire de la fabrication à la commande, ce qui permettrait de faire exactement les produits requis. Cependant le faible temps de mise à disposition ne le permet pas. La production pour stock, qui permettrait d'avoir un temps de mise à disposition quasi nul, n'est pas non plus envisageable, car cela nécessiterait de stocker un exemplaire de chaque produit fini et la diversité est telle que cela n'est pas réalisable.

Une solution intermédiaire consiste à concevoir l'offre produit comme une famille de produit avec des modules plus ou moins indépendants [2]. Ceci permet de fabriquer séparément chaque module

pour stock. Lors de la réception d'une commande client, il ne reste alors plus qu'à assembler le produit demandé à partir des modules en stock. Cette stratégie s'appuie avantageusement sur les concepts de la différenciation retardée [3]. Le temps de mise à disposition apparent des produits finis est alors le temps d'assemblage des modules nécessaires à la satisfaction de chaque commande. Ce temps sera appelé le *temps d'assemblage final*.

Une difficulté consiste alors à définir quelle est la composition du stock de modules qui permet d'assurer un assemblage final en un temps maximal imposé, cela au coût minimum. Plus précisément cette question revient à se demander combien de modules il est nécessaire de stocker, et quelle doit être la composition de chaque module.

La section 2 précise le problème traité et le positionnement du problème dans la littérature. La section 3 décrit la formalisation adoptée. La section 4 décrit la manière dont est utilisé le recuit simulé pour ce problème. La section 5 présente les résultats obtenus. Enfin la section 6 conclue ce travail.

## 2 Description du problème traité / Positionnement du problème

Les travaux proposés ici s'inspirent d'un cas industriel réel, dans l'automobile. Un sous-traitant reçoit la composition exacte de chaque produit à livrer un temps $X$ avant l'instant de livraison. Cette livraison doit être effectuée, en synchrone, en bord de la chaîne d'assemblage du donneur d'ordres – le constructeur automobile.

La diversité des modèles de véhicules proposés par le donneur d'ordres est très nettement supérieure au nombre de véhicules assemblés en une année [4]. Une production pour stock n'est donc pas envisageable. Inversement le temps de fabrication de chacun des produits considérés est très nettement supérieur au délai imposé de mise à disposition $X$. Une production de type fabrication à la commande est donc impossible pour satisfaire ce délai.

La problématique du fournisseur est donc la suivante : comment proposer une diversité de produits quasi infinie avec un temps de mise à disposition (très) inférieur au temps de fabrication de chaque produit ? La stratégie adoptée est alors de s'appuyer sur un découpage modulaire du produit, suivi d'un assemblage à la commande. Chacun des modules sera fabriqué par anticipation et stocké pour être assemblés les uns aux autres afin de réaliser les produits finis demandés.

La conception modulaire a des implications importantes tout au long de la chaîne logistique, des fournisseurs jusqu'aux clients. De nombreux travaux permettent d'envisager la conception modulaire avec des points de vue différents. Notons les travaux de Kusiak et Huang [5, 6, 7] qui proposent des découpages modulaires des produits à partir de leur représentation matricielle produit/fonction et/ou produit/process en fonction des cas.

De nombreux travaux se sont adressés à la définition des sous assemblages. Agard et Kusiak [8] utilisent les techniques de data mining. Gupta et Krishnan [9] recherchent des sous assemblages génériques respectant les contraintes d'ordre d'assemblage. Hsuan [10] quant à elle s'intéresse aux sous-assemblages en considérant les implications sur la relation fournisseur – donneur d'ordres.

## 3 Formalisation

Considérons $n$ composants élémentaires $a_i$ $i \in [1, n]$ permettant de réaliser jusqu'à $2^n - 1$ produits finis différents.

L'étape de pré assemblage permet de réaliser $SV$ modules $m_i$, $i \in [1, SV]$. On appelle $|m_i|$ le nombre de composants dans le module $i$. Soit $C$ l'ensemble des modules : $C = \{m_1, \ldots, m_{SV}\}$.

Notons $AT(C)$ le temps d'assemblage final moyen pour une demande de produits finis $D$ à partir des modules de $C$. Notons $X$ le temps d'attente toléré par le client.

Le modèle proposé [11] s'exprime alors :

$$\underset{C}{Min}\ \text{Coût(C)} \tag{1}$$

$$\text{t.q. } AT(C) < X \tag{2}$$

avec

$$\text{Coût(C)} = \alpha \sum_{m_i \in C} \left( |m_i| - 1 \right) + \beta \sum_{m_i \in C} \left( |m_i| \right) + \gamma SV + \delta AT(C) \tag{3}$$

Tel que

$$\alpha \sum_{m_i \in C} \left( |m_i| - 1 \right) \tag{4}$$

L'équation 4, représente le coût de pré assemblage, celui-ci est considéré proportionnel au nombre d'étapes d'assemblages effectuées.

$$\beta \sum_{m_i \in C} |m_i| \tag{5}$$

Le coût de transport est proportionnel à la taille des modules transportés (équation 5).

$$\gamma\, SV \tag{6}$$

Le coût de stockage est proportionnel au nombre de modules stockés (équation 6).

$$\delta\, AT(C) \tag{7}$$

L'équation7 permet d'évaluer le coût d'assemblage final moyen. *AT(C)* représente le temps d'assemblage final moyen et est déterminé de la manière suivante :

$$AT(C) = \sum_{j=1}^{2^n - 1} ( p(P_j) \times NA(P_j, C)) \tag{8}$$

$p(P_j)$ représente la probabilité de demande du produit $P_j$ et $NA(P_j, C)$ est le nombre d'opérations d'assemblage nécessaires pour réaliser le produit $P_j$ à partir de la composition *C*.

Une instance du problème sera constituée d'une demande *D*, c'est-à-dire les probabilités $p(P_j)$ pour l'ensemble des $2^n - 1$ produits finis.

## 4 Recuit simulé

Depuis son introduction dans les années 1980 [12], le recuit simulé a démontré être une bonne solution pour un large panel de problèmes [13]. Ce type de méta-heuristiques a déjà été utilisé pour la conception de modules. Kreng et Lee [14] utilisent cette méthode pour prendre en compte les relations physiques et fonctionnelles entre composants.

### 4.1 Espace de recherche

L'espace de recherche est constitué de l'ensemble des compositions permettant d'assembler tous les produits finis sans standardisation. Si les $2^n - 1$ produits sont réalisables, cette contrainte permet de caractériser les compositions acceptables. Ces compositions doivent contenir, entre autre, les composants élémentaires $a_i$. Il existe donc $2^n - 1 - n$ modules potentiels qui peuvent ou non être présents dans la composition. Ce sont donc les $2^{2^n-1-n}$ compositions respectant cette contrainte qui constituent l'espace de recherche.

Notons que la taille de l'espace de recherche est rapidement trop importante pour une exploration exhaustive. De plus, la fonction objectif ne permet pas d'approche de type gradient. Ces différentes caractéristiques du problème nous ont orientés vers l'utilisation d'une méta-heuristique.

### 4.2 Fonction d'évaluation

Il n'est pas possible de déterminer *a priori* si une composition appartenant à l'espace de recherche garantie ou non que le temps d'assemblage moyen sera inférieur au temps d'attente toléré par le client.

Pour utiliser le recuit simulé en intégrant cette contrainte (eq. 2), la fonction d'évaluation utilisée dans l'algorithme diffèrera de la fonction objectif. Ainsi la fonction d'évaluation est définie comme suit :

$$Cout\_evalue(C) = \begin{cases} coût & si\ \delta AT(C) \le X \\ coût + penalite \times (\delta AT(C) - X) & sinon \end{cases}$$

Le coefficient *penalite* permettra de pénaliser les compositions ne vérifiant pas la contrainte. La pénalité est modélisée afin d'être proportionnelle à l'écart observé, ce qui permet de représenter le type de contrat passé entre le donneur d'ordres et le fournisseur étudiés.

### 4.3 Fonction de déplacement

La fonction de déplacement permet de générer uniquement des voisins appartenant à l'espace de recherche. Seront considérées comme voisines deux compositions distantes d'un produit. Explicitons cette notion de distance, deux compositions $C$ et $C'$ sont distantes d'un produit ssi :

$$\begin{cases} \exists\, j \in [[1, 2^n - 1]] \text{ tel que } \forall k \in [[1, 2^n - 1]] \backslash j \ \left(C[k] = C'[k] \text{ et } C'[j] = (1 - C[j])\right) \\ \text{Ou} \\ SV_C = SV_{C'} \end{cases}$$

L'algorithme suivant (**accepte**) donne le schéma de mouvement. Il explicite comment un voisin devient la composition actuelle.

Un voisin donnant un coût plus faible est toujours accepté. Un voisin générant un coût plus élevé a une probabilité non nulle d'être accepté. Cette probabilité est liée au facteur *temperature* (Cf. 4.4).

ALGORITHME **accepte**

ENTREE : augmentation $\delta coûts$, temperature $T$

SORTIE : booléen

SI ($\delta coûts < 0$)

    RETOURNE *VRAI*

SINON

    $A \leftarrow e^{-(\delta coûts/T)}$ // *le degré de mouvements ascendants*

    $d \leftarrow U[0,1]$

    SI ($d < A$)

        RETOURNE *VRAI*

    SINON

        RETOURNE *FAUX*

    FIN SI

FIN SI

FIN **accepte**

### 4.4 Évolution de la température

Afin de garantir que l'algorithme de recuit converge, il faut s'assurer que la température décroît et qu'elle tend vers zéro. L'algorithme suivant opère une réduction de la température permettant de garantir cette propriété.

ALGORTIHME **nouvelletemperature**

ENTRÉE : temperature actuelle $T$

SORTIE : nouvelle temperature *newT*

$newT \leftarrow T \times coefficient\_de\_refroidissement$

RETOURNE *newT*

FIN **nouvelletemperature**

### 4.5 Implémentation

L'algorithme de recuit simulé utilise les fonctions définies précédemment. Il permet à partir d'une composition $C_0$ tirée au hasard d'obtenir une solution garantissant la contrainte de temps (eq. (2)). Le coût minimal trouvé au cours des itérations est retourné en fin d'algorithme.

ALGORITHME **recuit simule**
ENTRÉE : solution initiale $C_0$, # *iterations*
SORTIE : meilleure solution *coûts*$^*$
$C \leftarrow C_0$
$T \leftarrow T_0$
*coûts*$^*$ ← *coût_evalue(C)*
$m \leftarrow 0$
TANT QUE ($m <$ # *iterations*)
    $C_1 \leftarrow$ **voisin**($C$)
    $\delta$*coûts* ← *coût_evalue* ($C_1$) - *coût_evalue* ($C$)
    SI(**accepte**($\delta$*coûts*, $T$)
        $C \leftarrow C_1$
    FIN SI
    $T \leftarrow$ **nouvelletemperature**($T$)
    $m \leftarrow m+1$
    SI (*coût_evalue* ($C$) < *coûts*$^*$)
        *coûts*$^*$ ← *coûts*($C$)
    FIN SI
FIN TANT QUE
RETOURNE *coûts*$^*$
FIN **recuit simule**

## 5 Résolution

Les exemples présentés dans ce paragraphe ont les caractéristiques suivantes : la fonction objectif est paramétrée avec $\alpha = 2$, $\beta = 0.2$, $\gamma = 0.8$, $\delta = 20$, le coefficient de pénalité est fixé à 1000 et les demandes sont de types homogènes [11]. Le coefficient de refroidissement est de 0,5.

### 5.1 Recuit simulé

La Figure 1 représente les coûts des compositions obtenues par l'application du recuit simulé sur une instance. Ces coûts sont donnés en fonction du nombre d'itérations réalisées. L'axe secondaire présente la taille de stock de ces mêmes compositions. Les 20 premières itérations présentent un coût très élevé, cela traduit le dépassement de la limite de temps d'assemblage autorisé ($X$).

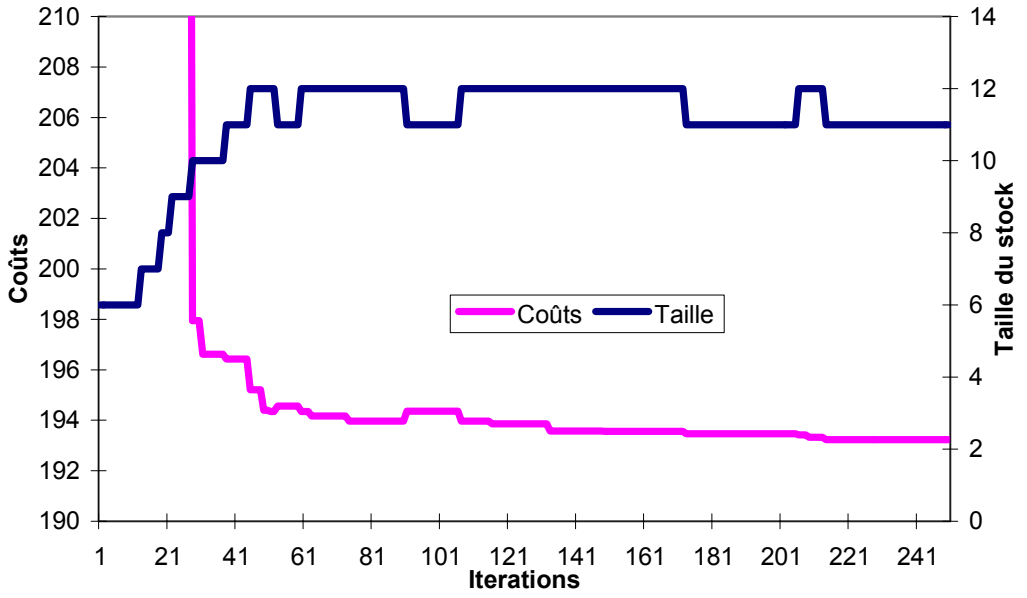


Figure 1 : Recuit simulé sur une instance.

Après 150 itérations, le coût ne varie plus qu'infimement tandis que la taille du stock évolue encore. Étudions ce phénomène. La Figure 2 présente l'évolution comparée de la taille de stock et du temps moyen d'assemblage sur la même instance. La stabilisation du coût après la 150ème itération s'explique par une évolution opposée de la taille du stock et du temps d'assemblage associé.

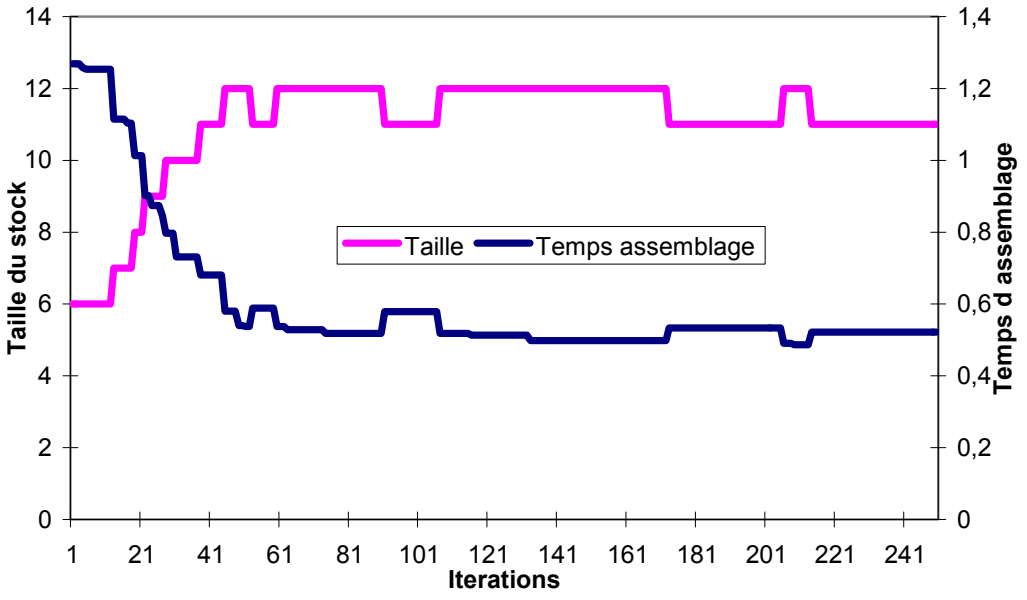


Figure 2 : Évolution de la taille du stock et du temps d'assemblage.

### 5.2 Résultats et analyses

Afin d'élargir les conclusions, l'algorithme de recuit simulé décrit précédemment a été utilisé sur 100 instances (100 répartitions des demandes en produits finis différentes). Les coûts moyens sont présentés en Figure 3.

Le traitement d'une instance de taille 256 (i.e. 256 produits différents) demande en moyenne 85 secondes (sur un Pentium® IV 3.00GHz).

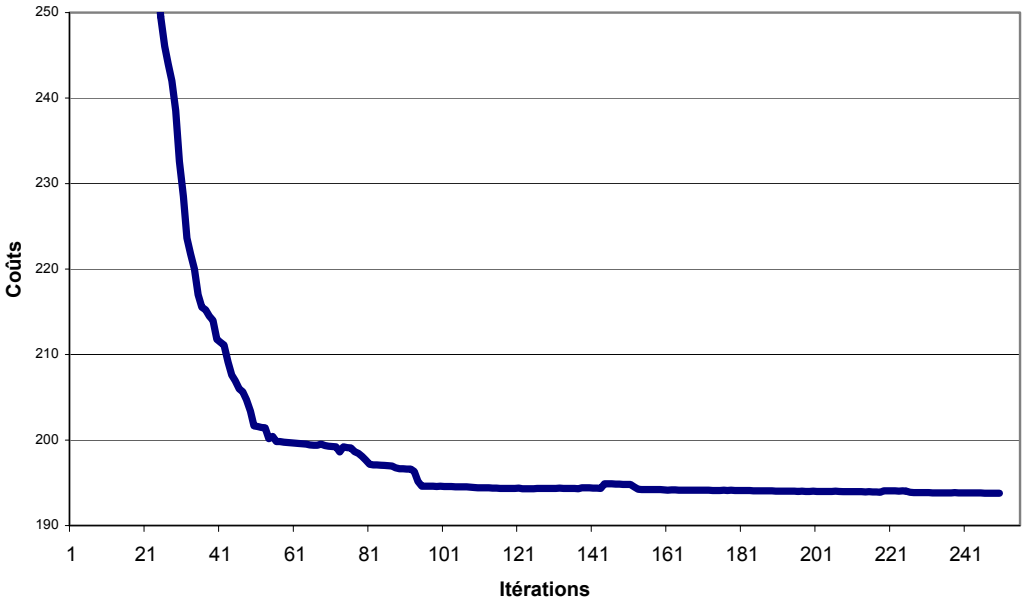


Figure 3 : Coûts obtenus par recuit simulé.

L'algorithme de recuit simulé a permis, pour toutes les instances étudiées, d'obtenir rapidement une solution, i.e. une composition de stock de produits semi-finis, garantissant le respect de la contrainte temporelle. Remarquons que plusieurs compositions différentes peuvent conduire aux mêmes coûts. Le choix entre ces compositions devra alors s'effectuer en fonction des orientations stratégiques de l'entreprise.

## 6 Conclusions

Le problème présenté visait à la constitution d'un stock de produits semi finis. Les modules considérés permettent de respecter un temps d'assemblage final moyen donné pour une demande. Le tout est réalisé en minimisant les coûts de la chaîne logistique par un recuit simulé. Les tests menés ont démontré la pertinence de la méthode choisie

Une poursuite de ce travail permettra d'étudier le poids relatif des coûts de main d'œuvre et des coûts de transports. Ceci permettra de mesurer l'influence des coûts de la chaîne logistique sur le nombre de modules et la constitution de ces modules. De même des considérations de dépendance vis à vis des sites délocalisés seront à évaluer.

**Références**


[1] C. LAMBEY-CHECCHIN, "Le sacrifice perçu : le cas d'acheteurs de voiture neuve", 2ème congrès International des tendances du Marketing en Europe, 2002

[2] B. AGARD, "Conception et fabrication des familles de produits : État de l'art", Journal Européen des Systèmes Automatisés, Vol. 38, No. 1-2, pp. 59-84, 2004.

[3] H. LEE, C. TANG, "Modeling the costs and benefits of delayed product differentiation", Management Science, Vol. 43, No. 1, p. 40-53, 1998.

[4] V. BERNIER, "Sur une nouvelle politique de gestion de flux : le cadencement reséquençable", Thèse de Doctorat, Institut National Polytechnique de Grenoble, 2000.

[5] A. KUSIAK, "Engineering Design: Products, Processes, and Systems", Academic Press, San Diego, CA, 1999.

[6] C. HUANG, A. KUSIAK, "Modularity in design of products and systems", IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 28, No. 1, pp. 66-77, 1998.

[7] A. KUSIAK, C. HUANG, "Development of modular products", IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part A, Vol. 19, No. 4, pp. 523-538, 1996.

[8] 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.

[9] S. GUPTA, V. KRISHNAN, "Product family-based assembly sequence design methodology", IIE Transactions, Vol. 30, No. 10, pp. 933-945, 1998.

[10] J. HSUAN, "Modularization in black-box design: implication for supplier-buyer partnerships", DRUID (Danish research unit for industrial dynamics) winter conference, 1998.

[11] B. AGARD, C. da CUNHA, (soumis) "Influence de la nature de la demande sur la constitution d'un stock de produits semi-finis", 6ème Congrès de Génie Industriel, Besançon, France, 2005.

[12] S. KIRKPATRICK, C. GELATT, M. VECCHI, "Optimization by simulated annealing", Science, Vol. 220, pp 671-680, 1983.

[13] R. OTTEN, L. van GINNEKEN, "The annealing algorithm", Kluwer academic publishers, 1989.

[14] V. KRENG, T-P LEE. "Modular product design with grouping genetic algorithm – a case study", Computers and industrial engineering, Vol. 46, No. 3, pp. 443-460, 2004.