HAL
open science

CONFIGURATION D'UN SYSTÈME D'ASSEMBLAGE MULTI-NIVEAUX SOUS INCERTITUDES DES DÉLAIS D'APPROVISIONNEMENT

Oussama Ben-Ammar, Hélène Marian, Alexandre Dolgui

To cite this version:

Oussama Ben-Ammar, Hélène Marian, Alexandre Dolgui. CONFIGURATION D'UN SYSTÈME D'ASSEMBLAGE MULTI-NIVEAUX SOUS INCERTITUDES DES DÉLAIS D'APPROVISIONNEMENT. 9th International Conference on Modeling, Optimization & SIMulation, Jun 2012, Bordeaux, France. 10p. hal-00728594

HAL Id: hal-00728594

https://hal.science/hal-00728594v1

Submitted on 30 Aug 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

9e Conférence Internationale de Modélisation, Optimisation et SIMulation - MOSIM'12
06 au 08 Juin 2012 - Bordeaux - France
« Performance, interopérabilité et sécurité pour le développement durable »

CONFIGURATION D'UN SYSTÈME D'ASSEMBLAGE MULTI-NIVEAUX SOUS INCERTITUDES DES DÉLAIS D'APPROVISIONNEMENT

Oussama BEN AMMAR, Hélène MARIAN, Alexandre DOLGUI

Institut Henri Fayol
École Nationale Supérieure des Mines de Saint Étienne
158, cours Fauriel, 42023 Saint-Étienne Cedex 2, France
obenammar@emse.fr, marian@emse.fr, dolgui@emse.fr

RESUME : *Pour l'entreprise, la gestion des stocks est un véritable enjeu. Il faut pouvoir satisfaire les clients à moindre coût. Pour cela, il faut être en possession des composants nécessaires pour fabriquer les produits demandés à la date voulue. Nous nous intéressons à un problème d'assemblage multi-niveau. Notre étude a comme objectif le choix de paramètres des systèmes MRP lorsque les entreprises sont soumises aux incertitudes des délais de fabrication et d'approvisionnement. Nous proposons un modèle de simulation et un algorithme génétique pour l'analyse et l'optimisation des temps de cycle planifiés.*

MOTS-CLES : *Délais d'approvisionnement aléatoire, système d'assemblage, gestion de stocks, méthode MRP, simulation, optimisation.*

1 INTRODUCTION

Ces dernières années, le combat des entreprises pour réduire leurs coûts s'est accentué. Or les stocks représentent une immobilisation financière importante, et il est donc primordial de les maîtriser. En outre, une mauvaise gestion de stocks conduit soit à des retards de livraison, qui engendrent des frais, soit à des stocks inutiles. Ces derniers peuvent être créés à différents niveaux (des matières premières aux produits finis), coûtent de l'argent et immobilisent des fonds. C'est pourquoi, il faut chercher des méthodes de planification de la production et des approvisionnements efficaces pour savoir exactement, et cela pour chaque composant, quand commander et en quelle quantité.

Dans le contexte industriel actuel, l'offre est largement supérieure à la demande, et par conséquent les clients sont de plus en plus exigeants. Pour se distinguer, les entreprises doivent proposer à leurs clients des produits de bonne qualité, aux meilleurs coûts, et avec des délais maîtrisés et les plus courts possibles. Pour ce dernier point, les méthodes de planification de type MRP sont souvent utilisées en entreprises, elles s'appuient sur un système d'information bien développé permettant la gestion des données de production comme la demande, la production, les stocks, les délais d'approvisionnement, etc.

Toutefois, le système MRP est conçu pour un environnement stable. Or de nos jours, la chaîne logistique est devenu très vulnérable et loin d'être déterministe (Dolgui et Prodhon, 2007). L'entreprise doit s'adapter aux changements des conditions économiques (évolution des coûts, augmentation de prix des matières premières, etc.), aux problèmes techniques (pannes machines, capacité limitée, retard de transports, etc.), aux contraintes concurrentielles (rapport qualité/prix attractif, la veille technologique, choix des fournisseurs, etc.).

L'incertitude peut également perturber le processus de production (Koh et Saad, 2003). Dans l'article (Wazed et *al.*, 2009), les auteurs ont identifié les principaux facteurs et causes d'incertitude en environnement industriel: la demande, les délais d'approvisionnement et la quantité produite peuvent varier d'une manière aléatoire. Pour éviter les conséquences de ces incertitudes, les entreprises utilisent les stocks de sécurité et/ou surestiment les délais d'obtention (Buzacott et Shanthikumar, 1994). Mais ces deux solutions génèrent des coûts supplémentaires. Il est donc intéressant de développer des méthodes de planification de la production et des approvisionnements qui permettent de minimiser ces coûts.

Quelques publications récentes montrent comment les stocks de sécurité permettent d'entraver des différentes causes d'incertitude. Dans (Camisullis et Giard, 2010), les auteurs proposent une détermination des stocks de sécurité, combinant différentes sources d'aléas. La variabilité des délais d'approvisionnement est traitée par exemple par (Whybark et Williams, 1976); les auteurs ont montré que l'utilisation du délai de sécurité est plus efficace que l'utilisation de stock de sécurité quand les temps d'approvisionnements sont incertains.

Dans cet article, nous nous intéressons au cas où les délais d'approvisionnement sont incertains. Nous considérons une entreprise d'assemblage qui gère sa chaîne logistique multi-échelon avec l'approche MRP globale. Nous nous intéressons au problème de paramétrage du MRP sous incertitude des délais de fabrication et d'approvisionnement.

Nous commençons par un examen rapide des travaux antérieurs relatifs à l'optimisation d'un système d'assemblage multi-niveau sous incertitudes (section 2). Nous proposerons ensuite dans la section 3 un modèle de simulation permettant trouver une estimation du coût total moyen pour chaque solution envisagée. Dans la section 4 nous modifions le modèle mentionné dans (Hnaien et *al*, 2008) pour pouvoir comparer le coût total moyen trouvé par cette méthode et celui trouvé par notre modèle de simulation. Nous parlerons du couplage de ce modèle avec un algorithme d'optimisation dans la section 5. Un exemple et des premiers résultats seront énoncés dans la section 6 avant de conclure et de présenter quelques pistes de recherche future.

2 TRAVAUX ANTÉRIEURS

Les modèles avec des délais d'approvisionnement aléatoires dépendent de la structure de la chaîne logistique. Pour ce type de problème, plusieurs types de structures de chaîne logistique sont identifiés dans la littérature à savoir les systèmes simples (un seul fournisseur d'un seul type de composant), les chaînes d'approvisionnement à structure linéaire et les systèmes d'assemblage etc. Pour ces trois types de chaînes logistiques nous distinguons les structures à un niveau et celles à plusieurs niveaux. Nous nous intéressons aux systèmes d'assemblage multi-niveaux.

L'incertitude sur les délais d'approvisionnement pour les systèmes d'assemblage à un seul niveau sont traités, par exemple, par (Dolgui et Ould Louly, 2008) et (Ould Louly et *al.*, 2008). Ces auteurs ont supposé dans leurs études que les délais d'approvisionnement suivent une loi discrète et que la demande est fixe et connue. Leur objectif est de trouver les stocks initiaux en composants qui permettent de réduire le coût total moyen qui est égal à la somme du coût de stockage des composants, du coût de stockage du produit fini et du coût de rupture en produit fini. Les auteurs ont développé une procédure par séparation et évaluation (PSE) utilisant des propriétés de dominance et des bornes supérieures et inférieures de la fonction objectif afin de trouver la solution optimale.

Dans (Tang et Grubbström, 2003), les auteurs ont étudié un système d'assemblage à deux niveaux, les délais d'approvisionnement sont des variables aléatoires discrètes, et la demande en produit fini est connue et constante. Leur objectif était de minimiser la somme de coûts de rupture et de coûts de stockage du produit fini. Ils ont utilisé la transformation de Laplace pour trouver les dates optimales de lancement des ordres aux fournisseurs.

Le même problème a été étudié par (Dolgui et *al.*, 2008). Deux méthodes basées sur une procédure de PSE et un algorithme génétique ont été mises en œuvre dans le but de chercher les meilleures dates de lancement des ordres d'approvisionnement aux fournisseurs, afin de minimiser l'espérance mathématique de coût total.

Dans (Fallah-Jamshidi et *al*, 2011), les auteurs étudient le même problème, ils ne minimisent pas le coût total mais cherchent plutôt à trouver un bon compromis entre le coût de stockage et le coût de rupture. Ils proposent un algorithme génétique (l'algorithme « Electromagnetism like Mechanism ») ainsi qu'une procédure de recherche locale permettant d'accélérer la convergence de l'algorithme. Ces deux méthodes sont utilisées pour résoudre ce problème bi-objectif en déterminant l'ensemble des solutions optimales de Pareto. Une comparaison avec l'approche proposée dans (Hnaien et *al.*, 2009) est également présentée. En termes de qualité de résultats, l'algorithme proposé semble être plus performant qu'un algorithme génétique II.

Dans cet article, nous déterminons, pour un système d'assemblage à *n* niveaux, les dates de lancement d'ordres d'approvisionnement qui minimisent l'espérance mathématique de coût total. Notre méthode est basée sur le couplage entre un algorithme génétique et un modèle de simulation.

3 DÉSCRIPTION DU PROBLÈME

3.1 Modèle étudié

Nous nous intéressons à la planification de la production et des approvisionnements des composants pour l'assemblage d'un produit fini, effectué étape par étape, c'est-à-dire dans une chaîne logistique multi-niveau (voir Figure 1). Ces travaux se situent dans la continuité des travaux de (Dolgui et Ould Louly, 2008) et (Hnaien, 2008) et afin de nous rapprocher des méthodes industrielles de planification de type MRP, nous nous plaçons comme eux dans un environnement temporel discret. Le modèle est mono-période.

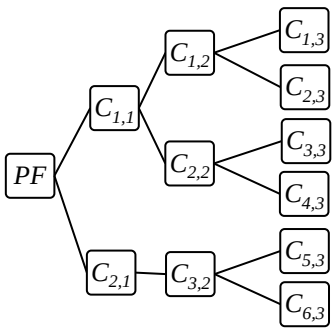

Figure 1 : Un système d'assemblage à 3 niveaux

Nous supposons que :

- Un produit fini est obtenu par l'assemblage de plusieurs types de composants différents, et ce, à plusieurs niveaux,
- Les délais d'obtention sont des variables aléatoires discrètes suivant des lois connues,
- Le temps d'assemblage est inclus dans le délai d'approvisionnement d'un composant (sauf pour les composants du dernier niveau de la nomenclature.), le délai d'obtention correspond à la somme du délai de livraison et du temps d'assemblage,
- La demande est connue,
- La capacité est infinie.

Nous utiliserons par la suite les notations suivantes :

- T date de livraison souhaitée par le client pour le produit fini,
- D demande en produits finis, valeur entière,
- l niveau dans la nomenclature, $l = 1, \ldots, m$,
- $N_{l,i+1}$ nombre de composants $c_{l,i+1} \in S_{l,i}$ nécessaires pour assembler un composant $c_{l,i}$,
- $S_{l,i}$ ensemble des composants de niveau i+1 nécessaires à l'assemblage d'un composant $c_{l,i}$,
- $L_{l,i}$ délai d'obtention des composants $c_{l,i}$,
- $h_{l,i}$ coût unitaire de stockage d'un composant $c_{l,i}$ par unité de temps,
- $x_{l,i}$ délai d'approvisionnement planifié pour les composants de type $c_{l,i}$,
- b coût unitaire de retard de début d'assemblage d'un produit fini par unité de temps,
- r coût unitaire de stockage du produit fini par unité de temps,
- $X_{l,\,m}$ date de lancement du composant de type $c_{l,\,m}$ au dernier niveau m,
- $E[.]$ espérance mathématique.
- $(Z)^+$: Max(T, Z)
- $(Z)^-$: Min(T, Z)
- $M_{1,0} = \max_{i=1,..,N_1}(M_{i,1} + L_{i,1})$

Nous supposerons dans ce modèle que le flux est poussé. Ainsi, il suffit de choisir une date de lancement pour chacun des composants en bas de la nomenclature. Dès que tous les composants nécessaires sont disponibles, nous commençons l'assemblage de chacun des composants en remontant la nomenclature.

Notre but est alors de trouver les dates de lancement optimales des ordres aux fournisseurs pour les composants de plus bas niveau, afin de minimiser l'espérance du coût total. Ce dernier est composé: du coût de stockage des composants, du coût de stockage du produit fini et du coût de rupture en produit fini (voir Figure 2). Puisque nous sommes en milieu incertain, nous allons nous intéresser aux coûts moyens.

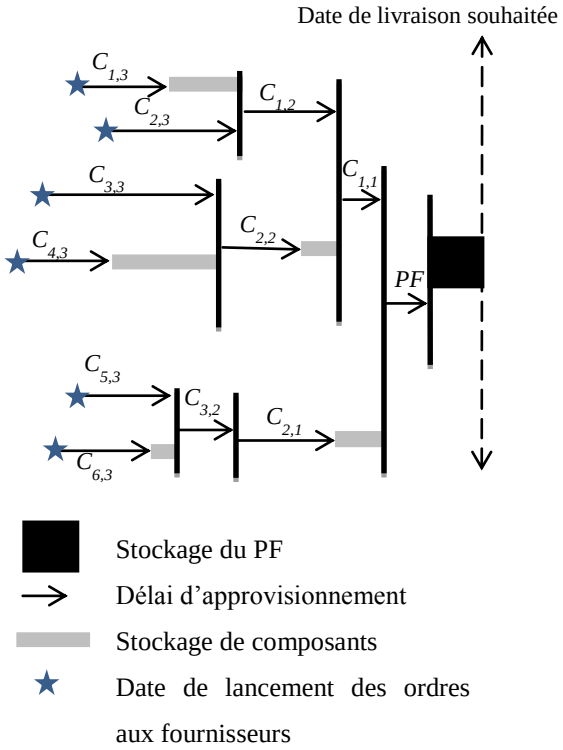

Figure 2 : Un système d'assemblage à 3 niveaux

Nous présentons ci-dessous le détail des différents coûts :

- Coût de stockage des composants $c_{l,i}$, $i = 2, \ldots, m-1$, sachant que l'arrivée des composants $c_{k,i+1} \in S_{l,i}$ nécessaires pour assembler un composant $c_{l,i}$, $l = 1, \ldots, N_i$ déclenche l'assemblage de ce composant:

$$C_{sc_2^{m-1}} = \sum_{i=2}^{m-1}\left(\sum_{l=1}^{N_{i-1}}\left(\sum_{c_{k,i}\in S_{l,i-1}} h_{k,i}\left(M_{l,i-1} - \left(M_{k,i} + L_{k,i}\right)\right)\right)\right)$$

$$\text{Où } M_{l,i-1} = \max_{c_{k,i+1}\in S_{l,i}}\left(M_{k,i} - L_{k,i}\right)$$

$$\text{et } M_{l,m} = \max_{c_{k,m}\in S_{l,m-1}}\left(L_{k,m} - X_{k,m}\right)$$

– Coût de stockage des composants $c_{l,1}$: Il y a un stockage des composants de type $c_{l,1}$, pendant le temps s'écoulant entre la date de leur arrivée sachant que l'arrivée des composants $c_{k,2} \in S_{l,1}$ nécessaires pour assembler un composant $c_{l,1}$, $l = 1, \dots, N_1$ déclenche l'assemblage de $c_{l,i}$:

$$C_{sc_1} = \sum_{l=1}^{N_1} \left(h_{l,1} \left(M_{1,0} - \left(M_{l,1} + L_{l,1} \right) \right) \right)$$

– Coût de stockage des composants $c_{l,m}$ du dernier niveau *m* de la nomenclature :

$$C_{sc_m} = \sum_{l=1}^{N_{m-1}} \left(\sum_{c_{k,m} \in S_{l,m-1}} h_{k,m} \left(M_{l,m-1} - \left(L_{k,m} - X_{k,m} \right) \right) \right)$$

– Coût de stockage du produit fini, si le produit est disponible avant la date de livraison prévue:

$$C_{sPF} = b.\left(T - min\left(M_{1,0}, T \right) \right)$$

– Coût de rupture en produit fini, si le produit est livré en retard par rapport à la date de livraison souhaité par le client :

$$C_{rPF} = r.\left(max\left(M_{1,0}, T \right) - T \right)$$

Les délais d'approvisionnement $L_{l,i}$ sont des variables aléatoires, le coût total *C(X, L)* est donc une variable aléatoire qui est égale à :

$$\begin{aligned} C(X,L) = {} & b.\left(T - min\left(M_{1,0}, T \right) \right) \\ & + r.\left(max\left(M_{1,0}, T \right) - T \right) \\ & + \sum_{l=1}^{N_{m-1}} \left(\sum_{c_{k,m} \in S_{l,m-1}} h_{k,m} \left(M_{l,m-1} - \left(L_{k,m} - X_{k,m} \right) \right) \right) \\ & + \sum_{i=2}^{m-1} \left(\sum_{l=1}^{N_{i-1}} \left(\sum_{c_{k,i} \in S_{l,i-1}} h_{k,i} \left(M_{l,i-1} - \left(M_{k,i} + L_{k,i} \right) \right) \right) \right) \\ & + \sum_{l=1}^{N_1} \left(h_{l,1} \left(M_{1,0} - \left(M_{l,1} + L_{l,1} \right) \right) \right) \end{aligned}$$

Notre objectif est de minimiser son espérance mathématique, c'est-à-dire le coût total moyen *E[C(X,L)]*.

3.2 Simulation

Dans les travaux antérieurs (Hnaien, 2009) et (Hnaien, 2008), le calcul exact du coût moyen n'a été obtenu que pour des cas particuliers de chaîne logistique à 2 niveaux de nomenclature. Les formulations mathématiques montrent que leur extension au cas plus général sera difficile. Nous avons donc développé un modèle permettant de calculer le coût total de stockage pour un système d'assemblage quelconque, quel que soit le nombre de niveaux. Notre modèle tient compte de la nomenclature, c'est-à-dire des relations de type (père/fils) entre tous les composants, ainsi que des lois discrètes du délai d'obtention pour chaque lien de nomenclature. Nous connaissons également les coûts de stockage unitaires de tous les composants et le coût de rupture pour le produit fini.

Ainsi pour les dates de lancement choisies, nous obtenons avec notre modèle de simulation le coût total pour une réalisation des lois de distribution. Or, nous voulons travailler à partir du coût moyen, il est donc nécessaire de faire plusieurs réplications d'une simulation.

3.3 Réplications

Un grand nombre de réplications indépendantes est indispensable pour obtenir une estimation du coût total moyen avec une précision donnée (taux de confiance, écart réduit). Pour trouver le nombre de réplications nécessaires pour obtenir un coût moyen stable, nous nous basons sur la théorie des probabilités ((Rubinstein et Y., 1981) et (Fishman, 1996)). Pour la suite, nous fixons le nombre de réplication à 2000 qui permet un d'obtenir un bon compromis entre la qualité des résultats obtenus et le temps de calcul pour un taux de confiance supérieur à 95%.

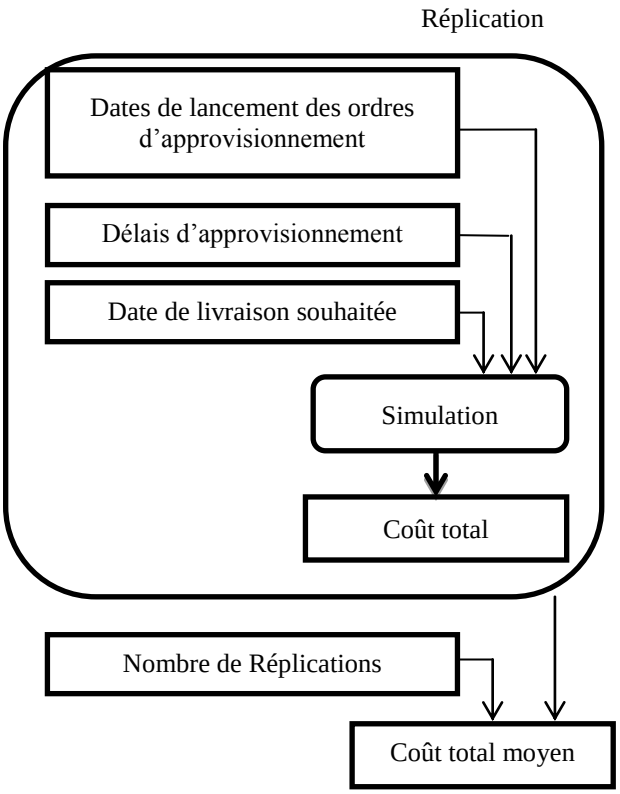

Figure 3 : Modèle de simulation

4 OPTIMISATION APPROCHÉE

Nous proposons une méthode d'optimisation approchée à base d'un algorithme génétique (AG), l'objectif est de déterminer les meilleures valeurs des dates de lancement des ordres aux fournisseurs tout en minimisant le coût total moyen $E[C(X,L)]$.

La première description du processus des algorithmes génétiques a été introduite par (Holland, 1975), puis (Goldberg, 1991) les a utilisés pour résoudre des problèmes concrets d'optimisation. Leurs champs d'application ont été appliqués avec succès sur plusieurs problèmes de génie industriel tels que le problème de gestion des stocks dans un système d'assemblage à deux niveaux face aux aléas des délais d'approvisionnement (Hnaien, 2010), le contrôle du temps d'approvisionnement dans un système d'assemblage à opérations multiples (Perkgoz, 2007), et la gestion de lotissements et des délais d'approvisionnement par (Kianfar, 2009).

Fonction Meilleur_sous_ensemble (A,n)
  Début
    retourne $S \subseteq A$, $|S| = n$ et $\forall s \in S$, $\nexists s' \in A \setminus S$, Fitness ($s'$) < Fitness($s$);
  Fin
  Ensemble_Population ← Population_initiale(N);
  Pour *i* de 1 à *Max génération* **faire**
    // *Sélection de reproduction*//;
    Ensemble_Parents ←Meilleur_sous_ensemble (Ensble_Population, $N/2$);
    // *Opérateurs de reproduction*//;
    Ensemble_Fils_Crois ← Croisement (Ensemble_Parents, P_{cross});
    Ensble_Fils_Mut ← Mutation (Ensenble_Fils_Crois, P_{mut});
    // *Sélection de remplacement*//;
    Ensemble_Population ← Meilleur_sous_ensemble (Ensemble_Parents ∪ Ensemble_Fils_Crois ∪ Ensemble_Fils_Mut, N);
  Fin

Figure 4 : Fonctionnement de l'algorithme génétique

Notre algorithme génétique commence par une génération d'une population initiale de N individus. Les meilleurs individus sont sélectionnés et ils seront manipulés par un opérateur de croisement qui les choisit selon une probabilité P_{cross}; et peuvent aussi être mutés par un opérateur de mutation avec une probabilité de mutation P_{mut}. Les phases de sélection et de recombinaison (croisement et mutation) permettent de générer deux nouvelles populations d'individus (une population après le croisement et une autre après la mutation), qui ont de bonnes chances d'être meilleurs que ceux de la génération précédente. Les individus issus de la phase de recombinaison seront insérés dans la nouvelle population; cette dernière sera évaluée en fonction de la valeur de la fonction coût $E[C(X,L)]$ de chacun de ses individus. De génération en génération, la force des individus augmente et un test d'arrêt sera effectué pour décider quand arrêter l'algorithme.
La figure 4 présente un schéma de fonctionnement général de l'algorithme génétique. Les différentes étapes de ce dernier seront présentées, en détail, dans les sections suivantes.

4.1 Codage

Le premier pas dans l'implantation d'un algorithme génétique est de créer la population d'individus initiaux. Chaque individu de la population est codé par un chromosome. Il représente un point dans l'espace de recherche. Toutes les solutions possibles au problème peuvent être codées à l'aide d'un chromosome. Dans notre d'étude, les variables de décision sont les dates de lancement de l'ordre d'approvisionnement à un fournisseur (variables de décision entières), donc chaque chromosome peut être codé avec un tableau de nombres entiers correspondant à nos variables de décisions.
Dans notre cas, chaque gène d'un chromosome représente une date de lancement d'ordre d'approvisionnement. La longueur du chromosome est égale à N_n qui correspond au nombre de composants au dernière niveau n de la nomenclature. La figure suivante présente un exemple des solutions pour le problème avec $N_n = 10$.

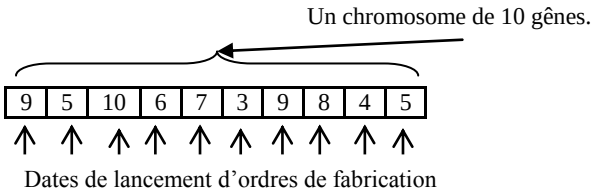

Figure 5 : Exemple d'un chromosome de 10 gènes

4.2 Génération de la population initiale

Dans cette étape, le problème principal est le choix de la taille et l'homogénéité de la population. Si la taille de la population est très grande, le temps de calcul augmente et demande un espace mémoire important. Dans le cas contraire, l'algorithme risque de ne pas avoir assez de variété dans la population. Il faut donc trouver le bon compromis pour générer des individus bien répartis dans l'espace de recherche, qui servira de base pour les générations futures.
Nous avons choisi de générer d'une manière aléatoire selon une distribution uniforme une population initiale de 60 individus. Le choix de cette population conditionne fortement la rapidité de convergence de l'algorithme. La position de l'optimum dans notre espace de résultat est totalement inconnue. Un test est fait avec la valeur de la fonction fitness pour éliminer les doublons et les remplacer par d'autres individus.

4.3 Évaluation : fitness

A chaque solution, on associe une fonction d'évaluation appelée une fonction "fitness" reliée à la valeur de la fonction «objectif». Cette fonction de performance décrit le mérite de l'individu représenté par des données contenues dans un chromosome. Dans notre cas la valeur numérique attribuée à un individu X_i est le coût total moyen $E[C(X_i)]$.
A chaque génération, une sélection d'individus est faite pour la reproduction par les opérateurs de croisement et de mutation. Chaque chromosome évalué par la fonction fitness et qui ne donne pas une bonne solution se verra attribuer une mauvaise valeur de la fonction "fitness" et aura une probabilité forte d'être éliminé par le processus de sélection.
Notre coût total moyen $E[C(X_i)]$ est calculé avec un taux d'erreur égal à 5 %. Après chaque évaluation par la fonction fitness, les doublons sont supprimés et remplacés par d'autres chromosomes issus de la population totale.

4.4 Stratégie de sélection

Après la création et l'évaluation de la population initiale, certains individus sont sélectionnés pour participer à la création de la nouvelle génération. Nous optons pour la méthode de sélection élitiste. Les individus sont sélectionnés pour se reproduire. Un individu est sélectionné selon sa performance. Dans notre cas, il existe deux phases de sélection:

- La sélection de reproduction: les $N/2$-individus les plus performants parmi la population initiale sont favorisés, ils subissent les opérations de reproduction (croisement et mutation). La sélection est $N/2$-élitiste, elle consiste à sélectionner les $N/2$ meilleures solutions.

- La sélection de remplacement: la sélection de la nouvelle population d'une génération à une autre. A chaque génération, la taille de la population est fixée à N. les individus de cette population sont choisis parmi $2N$ (N parents de la population courante, $N/2$ enfants obtenus par croisement et $N/2$ obtenus par mutations des fils issus du croisement.

A noter que les individus doublons sont supprimés à cette étape pour maintenir une diversité suffisante de la population.

4.5 Croisement

Dans les AGs, le croisement est considéré comme le principal opérateur pour produire de nouveaux individus et favoriser la diversification de la population. Un opérateur de croisement peut former deux nouveaux individus (fils) à partir de deux individus (parents). La création d'un couple d'enfants à partir d'un couple des parents est effectuée avec une probabilité de croisement P_{cross}, l'idée principale est d'échanger les gênes des parents afin de donner des enfants qui portent des propriétés combinées.

Parmi ces processus, nous pouvons citer le croisement multipoints. Un parent est composé de N_2 gênes, donc il est possible de le couper en n ($n = 1, \dots, N_i - 1$). Nous choisissons un opérateur de croisement à un seul point. L'ensemble des $N/2$ parents sélectionnés est partitionné de manière aléatoire en $N/4$ couples. Le point de croisement est choisi de manière aléatoire à chaque génération de l'algorithme dans l'intervalle $[1, \dots, N_i - 1]$. Chaque couple subit une opération de croisement avec une probabilité de croisement P_{cross}. Les couples qui ne subissent pas du croisement sont maintenus.
Soit un couple de parents $X^{p1} = \left(X_{1,2}^{p1}, \dots, X_{i,2}^{p1}, \dots X_{N_{2,2}}^{p1}\right)$ et $X^{p2} = \left(X_{1,2}^{p2}, \dots, X_{i,2}^{p2}, \dots X_{N_{2,2}}^{p2}\right)$ qui subit un croisement à un point j avec $j \in [1, \dots, N_2 - 1]$. Les deux fils $X^{f1} = \left(X_{1,2}^{f1}, \dots, X_{i,2}^{f1}, \dots X_{N_{2,2}}^{f1}\right)$ et $X^{f2} = \left(X_{1,2}^{f2}, \dots, X_{i,2}^{f2}, \dots X_{N_{2,2}}^{f2}\right)$ sont obtenus de la manière suivante :

$$\begin{cases} X_{i2}^{fk} = X_{i,2}^{pk}, i \in [1,2, \dots, j] \; et \; k \in \{1,2\} \\ X_{i2}^{f2} = X_{i,2}^{p2}, i \in [j+1, \dots, N_2] \\ X_{i2}^{f2} = X_{i,2}^{p1}, i \in [j+1, \dots, N_2] \end{cases}$$

Dans la figure suivante, un exemple d'opérateur de croisement, où N_2 la taille du chromosome est égale à 10 et le croisement est au point 3:

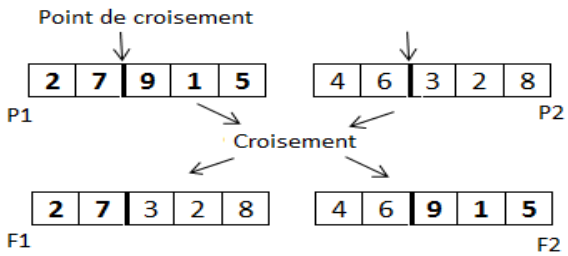

Figure 6: Exemple de croisement d'un couple de parents

4.6 Mutation

L'opérateur de mutation apporte à l'algorithme génétique une diversification nécessaire pour une exploration efficace de l'espace de recherche. Il permet d'atteindre la plupart des sous-espaces de solutions réalisables. En effet, la mutation joue le rôle d'un bruit et empêche l'évolution de se figer. Les propriétés de convergence d'un algorithme génétique sont fortement dépendantes de l'opérateur de mutation. Cet opérateur est utilisé avec une probabilité de mutation P_{mut}, généralement choisie faible (15%).
Dans notre étude, un individu $X = (X_{1,2}, \dots, X_{i,2}, \dots, X_{N_2,2})$ a une probabilité P_{mut} d'être muté. La mutation concerne uniquement un gène $X_{i,2}$ du chromosome. Ce gène est sélectionné aléatoirement à chaque génération par une simple distribution uniforme:

$$\begin{cases} X_{i,2}^{New} = X_{i,2}^{Old} + rand(-2,2) \\ X_{j,2}^{New} = X_{j,2}^{Old}, \forall j \in [1, \dots, N_2], j \neq i \end{cases}$$

4.7 Critère d'arrêt

Nous arrêtons l'algorithme après 500 générations. Nous avons remarqué que l'espace de recherche est suffisamment exploré et que la population n'évolue plus rapidement.

5 ÉTUDE EXPERIMENTALE ET RESULTATS

Afin de vérifier l'efficacité de notre algorithme, nous commençons par introduire le modèle de comparaisons. Ensuite nous présentons un système d'assemblage à deux niveaux avec 10 composants au dernier niveau. Dans l'étape suivante nous analysons les résultats obtenus sur des systèmes d'assemblages à deux niveaux, pour des instances inférieures à 60 composants au 2ème niveau. Enfin, nous présentons un système d'assemblage à 4 niveaux.
Pour la comparaison, nous utilisons la bibliothèque d'instances utilisée dans (Hnaien, 2008). Pour comparer les deux méthodes mises en œuvre, la mesure de performance utilisée est le coût total moyen obtenue avec la meilleure solution connue. Dans la première partie, nous fixons le nombre de réplications à 10000 et ensuite à 2000. La taille de la population *N* est égale à 60 individus, la probabilité de croisement est égale à 80%, la probabilité de mutation est égale à 15% et la condition d'arrêt est fixée à 500 générations.

5.1 Modèle de comparaison

Le modèle analytique que nous utilisons pour des comparaisons est proposé dans (Hnaien, 2008). Le coût total est la somme du coût de stockage de tous les composants, et le coût de stockage du produit fini. Dans ce modèle, nous rajoutons le coût de rupture du produit fini pour pouvoir comparer le coût total moyen obtenu d'une manière analytique à celui calculé par simulation.

Le coût total pour un système d'assemblage à deux niveaux s'exprime comme suit:

$$C(X,L) = \sum_{i=1}^{N_i}\left(\sum_{c_{k,2}\in S_{i,1}} h_{k,2}\left(M_{i,1} - \left(L_{k,2} - X_{k,2}\right)\right)\right) + \sum_{i=1}^{N_1}\left(h_{i,1}\left(\max_{i=1,\dots,N_1}\left(M_{i,1}+L_{i,1}\right) - \left(M_{i,1}+L_{i,1}\right)\right)\right) + r.\max_{i=1,\dots,N_1}\left(M_{i,1}+L_{i,1}\right)^{+} + b.\min_{i=1,\dots,N_1}\left(M_{i,1}+L_{i,1}\right)^{+}$$

$$\text{Où}\begin{cases} L = (L_{1,1},\dots,L_{N_1,1},\dots,L_{1,2},\dots,L_{N_2,2}) \\ X = (X_{1,2},\dots,X_{i,2},\dots,X_{N_2,2}) \\ M_{i,1} = \max_{c_{k,2}\in S_{l,1}}\left(L_{k,2} - X_{k,2}\right) \end{cases}$$

Le coût total moyen s'exprime comme suit:

$$\begin{aligned} \mathrm{E}\big(\mathrm{C}(\mathrm{X},\mathrm{L})\big) = {} & +\sum_{i=1}^{N_i} H_i\left(\sum_{s\geq 0}\left(1 - \prod_{c_{k,2}\in S_{i,1}} F_{k,2}\left(X_{k,2}+s\right)\right)\right) \\ & -\sum_{i=1}^{N_i} H_i\left(\sum_{s\geq 0}\left(\prod_{c_{k,2}\in S_{i,1}} F_{k,2}\left(X_{k,2}-s-1\right)\right)\right) \\ & +\mathrm{B}.\left(\sum_{s\geq 0}\left(\prod_{\mathrm{i}=1}^{\mathrm{N}_1}\left(\sum_{\mathrm{o}_1+\mathrm{o}_2=\mathrm{s}} \mathrm{P}\left(\mathrm{L}_{\mathrm{i},1}=\mathrm{o}_1\right).\prod_{\mathrm{c}_{\mathrm{k},2}\in \mathrm{S}_{\mathrm{i},1}} \mathrm{F}_{\mathrm{k},2}\left(\mathrm{X}_{\mathrm{k},2}-\mathrm{o}_2-1\right)\right)\right)\right) \\ & +R.\left(\sum_{s\geq 0}\left(1-\prod_{i=1}^{N_1}\left(\sum_{o_1+o_2=s} P\left(L_{i,1}=o_1\right).\prod_{c_{k,2}\in S_{i,1}} F_{k,2}\left(X_{k,2}+o_2\right)\right)\right)\right) \\ & -\sum_{i=1}^{N_i}\left(\sum_{c_{k,2}\in S_{i,1}} h_{i,2}\left(E\left(L_{k,2}-X_{k,2}\right)\right)\right) - \sum_{i=1}^{N_i}\left(h_{i,1}\left(E\left(L_{i,1}\right)\right)\right) \end{aligned}$$

$$\text{Où}\begin{cases} H_i = \sum_{c_{k,2}\in S_{i,1}} h_{k,2} - h_{i,1}, \forall\, i \in [1,\dots,N_1] \\ B = b + \sum_{i=1}^{N_i}\left(h_{i,1}\right) \\ R = r - \sum_{i=1}^{N_i}\left(h_{i,1}\right) \end{cases}$$

L'algorithme génétique est développé pour réduire le coût total moyen et chercher les meilleures dates de lancement des ordres $X_{k,2}, k = 1,\dots,N_2$, qui minimisent l'espérance mathématique de coût total $E(C(X,L))$.

5.2 Étude numérique

Nous effectuons un test sur un système d'assemblage à deux niveaux avec 2 types de composants au niveau 1 et avec 10 types composants au niveau 2 :

Nous avons fixé :
- La demande *D* en produits finis est fixée à 1,
- Le coût unitaire de stockage du produit fini est égal à 140,
- Le coût unitaire de retard de début d'assemblage d'un produit fini est 140,
- La date de livraison souhaitée par le client *T*=10,
- Les coûts unitaires de stockage $h_{i,1}$ des composants *i* au niveau 1 sont présentés dans le tableau 1,
- Les coûts unitaires de stockage $h_{i,2}$ des composants *i* au niveau 2 sont présentés dans le tableau 2.

i	**1**	**2**
$h_{i,1}$	40	100

Tableau 1 : Les coûts unitaires de stockage des $h_{i,1}$.

i	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
$h_{i,2}$	10	20	10	25	20	15	15	25	25	20

Tableau 2 : Les coûts unitaires de stockage des $h_{i,2}$.

Les délais d'approvisionnement $L_{l,i}$ des composants au niveau *i* de la nomenclature sont des variables aléatoires compris entre 1 et 5. La distribution cumulée des probabilités est donnée dans le tableau 3.

w	1	2	3	4	5
$P(L_{1,1}=w)$	0,435	0,749	0,935	0,944	1
$P(L_{2,1}=w)$	0,173	0,242	0,321	0,789	1
$P(L_{1,2}=w)$	0,114	0,119	0,127	0,49	1
$P(L_{2,2}=w)$	0,219	0,45	0,682	0,746	1
$P(L_{3,2}=w)$	0,2	0,357	0,382	0,652	1
$P(L_{4,2}=w)$	0,249	0,41	0,503	0,775	1
$P(L_{5,2}=w)$	0,316	0,622	0,8	0,847	1
$P(L_{6,2}=w)$	0,08	0,375	0,446	0,712	1
$P(L_{7,2}=w)$	0,305	0,611	0,798	0,919	1
$P(L_{8,2}=w)$	0,182	0,698	0,713	0,943	1
$P(L_{9,2}=w)$	0,35	0,379	0,383	0,858	1
$P(L_{10,2}=w)$	0,088	0,276	0,498	0,767	1

Tableau 3 : La probabilité de distribution des délais d'approvisionnement $L_{l,i}$.

Dans un premier temps, afin de permettre une comparaison juste, notre première expérience consiste à vérifier la validité de notre modèle (Figure 7). Nous réalisons 100 lancements du modèle de simulation et comparons le coût total moyen trouvé par simulation au coût total moyen exact trouvé par la modélisation analytique. Le coût total moyen trouvé par simulation est bien à plus que 95% dans l'intervalle de confiance ([513.40 ;508.23]) qui a été fixé d'avance.

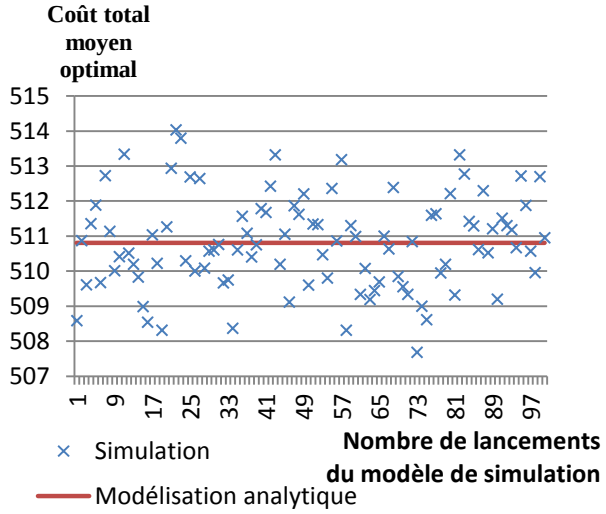

Figure 7 : Validation du modèle de simulation

Dans un deuxième temps, nous proposons un algorithme génétique partant de la même population initiale avec les mêmes paramètres que ceux utilisés dans (Hnaien, 2008). Nous comparons, le coût total moyen optimal calculé par la méthode analytique et par la simulation (voir Figure 8 et 9).

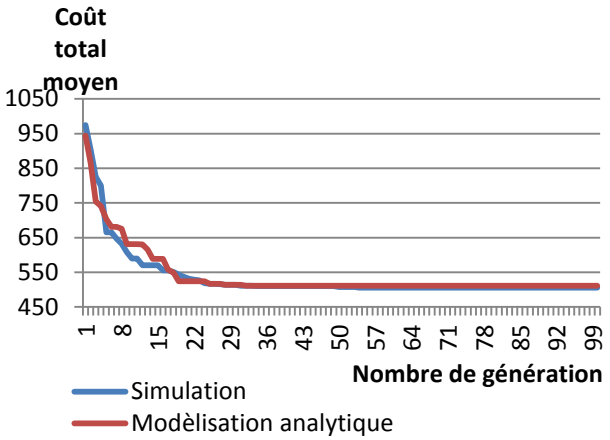

Figure 8: La convergence d'une population de 10 individus

La rapidité de la convergence est déterminée en fonction de la taille des chromosomes des individus et du nombre de générations. Une population dont les chromosomes sont de 10 gênes, convergent en moyenne au bout d'une trentaine de génération (voir figure 7), alors que, par exemple celle dont les chromosomes contiennent 50 gênes, convergent en moyenne au bout de 200 générations (voir figure 8). Donc la taille de la population initiale ainsi que le nombre de gênes d'un chromosome conditionnent la rapidité de l'algorithme.

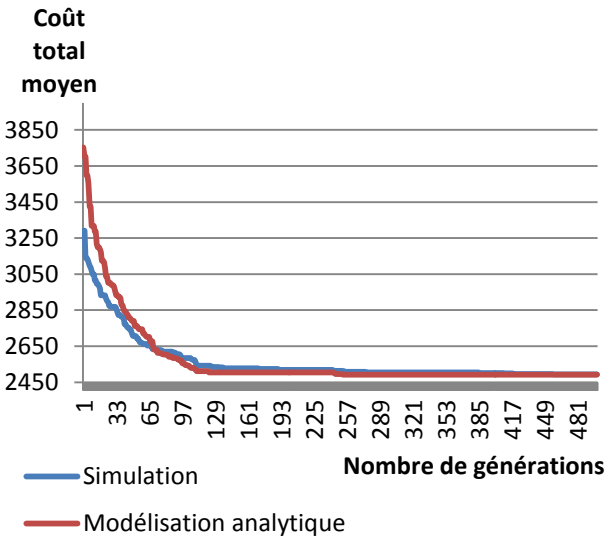

Figure 9 : La convergence d'une population de 50 individus

La meilleure solution connue est sélectionnée parmi les solutions obtenues du couplage de l'algorithme génétique et l'heuristique proposée par (Hnaien, 2008). L'écart entre les meilleurs et les pires runs (issus du couplage de l'algorithme génétique et le modèle de simulation) pour cette instance semble également très faible : 0,77% en moyenne et 1,57% pour le pire des cas (voir Figure 10).

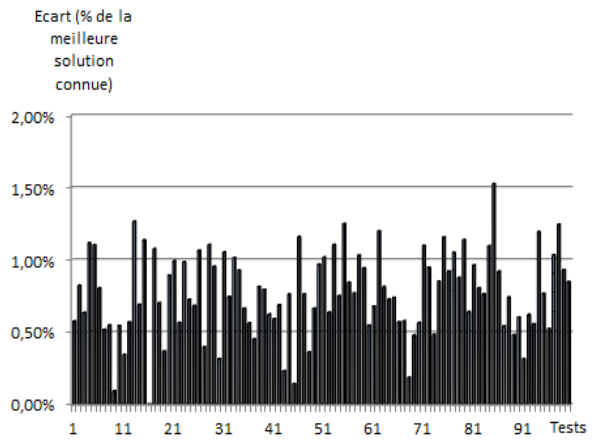

Figure 10 : L'écart entre la meilleure et le pire lancement du modèle de simulation de chaque instance

Si les lancements du modèle de simulation extrêmes (10%) ne sont pas considérés (les meilleurs 5% et les pires 5%), alors cet écart est très faible: 0,24% en moyenne et 0,46% pour le pire lancement du modèle de simulation.

Pour les instances de taille comprise entre 20 et 60, l'écart entre les meilleurs et les pires lancements du modèle de simulation est de 2% en moyenne (voir Figure 11). Les résultats montrent l'efficacité de l'algorithme proposé pour les problèmes de petite taille et de taille moyenne. La résolution optimale du problème dépend fortement du nombre de réplications et du nombre de générations.

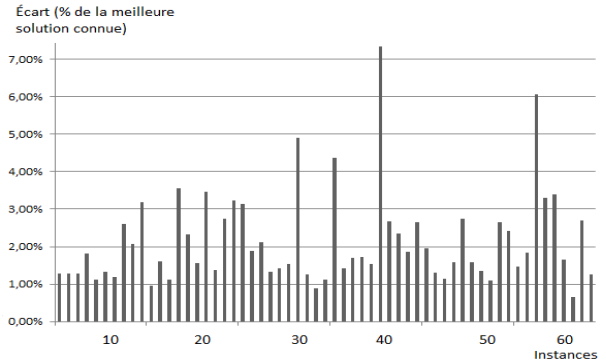

Figure 11: L'écart entre la meilleure et le pire run de chaque instance

Concernant le temps de calculs, il est important de noter que le couplage de la simulation avec un algorithme génétique est plus gourmand en temps de calcul que le couplage de la modélisation analytique avec un algorithme génétique (voir tableau 4). Cependant, ce couplage (simulation avec algorithme génétique) nous permet d'étudier des systèmes d'assemblage à plus de deux niveaux.

N_2	**10**	**20**	**30**
Simulation	67.63	117.62	163.23
Modélisation analytique	1,83	3,38	4,92

N_2	40	50	60
Simulation	217,58	274,39	330,74
Modélisation analytique	6,39	7,68	9,26

Tableau 4: Le temps moyen de calcul

6 CONCLUSION ET PERSPECTIVES

Nous nous sommes intéressés à la planification des réapprovisionnements des systèmes d'assemblage à plusieurs niveaux avec l'approche MRP quand les délais d'approvisionnement des composants sont aléatoires.

Nous proposons une nouvelle méthode d'optimisation pour les systèmes d'assemblage à plusieurs niveaux. Le critère utilisé est la somme du coût de stockage des composants, du coût de retard en produit fini, et du coût de stockage des produits finis. Notre approche se base sur une technique d'optimisation avec des variables de décision entières, ce qui est plus adapté au contexte du MRP qu'une approche continue, plus souvent utilisée dans la littérature. Nous avons développé un modèle de simulation et un algorithme d'optimisation basé sur un algorithme génétique. Nos résultats sont comparés à ceux d'une méthode basée sur une modélisation analytique.

Dans notre future recherche, nous voulons travailler sur la fonction "fitness" pour réduire le temps de calcul. Nous pensons aussi à étendre le modèle et les techniques proposées au contexte multi-période. Nous allons nous appuyer sur les travaux de (Hnaien, 2008) qui portent sur des modèles de planification multi-périodes pour des systèmes d'assemblage à un seul niveau avec des délais d'approvisionnement aléatoires. Nous allons aussi étendre notre étude aux systèmes d'assemblage multi-niveau à plus qu'un type de produits finis où les nomenclatures se croisent (les différents produits finis utilisent les mêmes types de composants).

REFERENCES

Axsäter, S., 2006. *Inventory Control, 2nd Ed.*, Springer, France.

Buzacott, J. A., and Shanthikumar, J. G., 1994. Safety stock versus safety time in MRP controlled production systems. *Management Science*, 40(12), p. 1678-1689.

Banks J., Carson II J. S., Nelson B. L., and Nicol D. M, 2010, *Discrete-Event System Simulation*, 5*ème* Ed., Pearson.

Camisullis, C. and Giard, V. (2010). Détermination des stocks de sécurité dans une chaîne logistique-amont dédiée à une production de masse de produits fortement diversifiés. *Journal Européen des Systèmes Automatisés*, 44(8), pp.975-1010.

Chiu H., and Huang., 2003. A multi-echelon integrated JIT inventory model using the time buffer and emergency borrowing policies to deal with random delivery lead times. *International Journal of Production Research*, 41, p. 475-490.

Dolgui, A., Ould Louly M.A., and Hnaien F., 2008. Supply planning for single-level assembly system with stochastic component delivery times and service level constraint. *International Journal of Production Economics*, 115(1), p. 236-247.

Dolgui A., Hnaien F., Marian H., and Ould Louly M.A., 2007. Planning order release dates for multilevel linear supply chain with random lead times. *Systems Science*, 31(1), p.19-25.

Dolgui A., and Prodhon, C., 2007. Supply planning under uncertainties in MRP environments: A state of the art. *Annual Reviews in Control*, 31, p. 269-279.

Dolgui A., and Ould Louly M.A., 2002. A model for supply planning under lead time uncertainty. *International Journal of Production Economics*,78, p. 145-152.

Fallah-Jamshidi S., Karimi N., and Zandieh M., 2011. A hybrid multi-objective genetic algorithm for planning order release date in two level assembly system with random lead times. *Expert Systems with Applications*, 38(11), p. 13549-13554.

Fishman G.S., 1996, *Monte-Carlo: Concepts, algorithms and applications*, Springer-Verlag.

Giard V., 2003. *Gestion de la Production et des flux*, Economica, Paris.

Guide V.D.R., and Srivastava R., 2000. A review of techniques for buffering against uncertainty with MRP systems. *International Journal of Production Planning and Control*, 11, p. 223- 233.

Hnaien F., 2008. *Gestion des stocks dans des chaînes logistiques face aux aléas des délais d'approvisionnements*. Thèse de doctorat, École Nationale Supérieure des Mines de Saint-Étienne, France.

Koh S.C.L., and Saad S.M., 2003. A holistic approach to diagnose uncertainty in ERP controlled manufacturing shop floor. *International Journal of Production Planning and Control*, 14 (3), p. 273-289.

Metropolis W., Rosenbluth A., Rosenbluth M., Teller A., and Teller E., 1953. Equation of the state calculations by fast computing machines. *Journal of Chemical Physic*, 21, p. 1087-1092.

Ould Louly M.A., 2001. *Optimisation des stocks des composants et planification de leurs réapprovisionnements pour les systèmes d'assemblage soumis à des aléas*. Thèse de doctorat, Université de Technologie de Troyes, France.

Rubinstein Y., 1981 *Simulation and the Monte Carlo method*. John Wileys and Sons, New York.

Tang O., and Grubbstrom R.W., 2003. The detailed coordination problem in a two-level assembly system with stochastic lead times. *International Journal of Production Economics*, 81-82, p. 415-429.

Whybark D. C., and Williams J.G., 1976. Material requirements planning under uncertainty. *Decision Science*, 7, p. 595-606.

Wazed M., Shamsuddin A. and Yusoff N. 2009. Uncertainty Factors in Real Manufacturing Environment. *Australian Journal of Basic and Applied Sciences*, 3(2), p. 342-351.