

METAHEURISTIQUES POUR LE CAPACITATED LOT SIZING PROBLEM

David LEMOINE, Sylvie NORRE

LIMOS CNRS UMR 6158

Antenne IUT de Montluçon

Avenue Aristide Briand

03101 Montluçon - France

{lemoine,norre}@monlut.univ-bpclermont.fr

Michel GOURGAND

LIMOS CNRS UMR 6158

Université Blaise Pascal

BP 10125

63173 Aubière Cedex – France

gourgand@isima.fr

RESUME : *L'élaboration du plan directeur de production joue un rôle majeur dans la planification de la production. Parmi les modèles de lot-sizing, le « Capacitated Lot Sizing Problem » (CLSP) est dédié à cette problématique. Une littérature abondante est consacrée à la résolution de ce problème, cependant peu de méthodes approchées de type métaheuristique ont été développées pour tenter de le résoudre. Dans cet article, nous proposons d'utiliser des métaheuristicques à base de recuit simulé. Le codage de la solution correspond au planning de production. Nous définissons un nouveau système de voisinages composé de plusieurs voisinages consistant à déplacer ou échanger des quantités produites dans le but de minimiser les coûts de production (lancement de campagne, stockage). Nous proposons également une fonction bi-objectif intégrant les coûts de production et une évaluation du dépassement de capacité qui permet au système de voisinage de s'affranchir temporairement des contraintes de capacité. Les premiers résultats obtenus sont prometteurs car ils permettent d'améliorer significativement les résultats que nous avons obtenus précédemment.*

MOTS-CLES : *CLSP, métaheuristique, système de voisinages, fonction bi-objectif.*

1. INTRODUCTION

La gestion de production s'intéresse à l'ensemble des décisions à prendre pour qu'une entreprise puisse fabriquer des produits conformes à la demande, en temps voulu et à des coûts permettant d'assurer sa pérennité. Ainsi, elle vise à l'obtention d'un compromis entre la satisfaction de la demande client et les critères de production tels que les coûts, les délais et la qualité des articles fabriqués. Traditionnellement (Giard V., 2003), cet ensemble est divisé en trois niveaux décisionnels hiérarchisés : les niveaux stratégique (long terme : plus de 18 mois), tactique (moyen terme : 6 à 18 mois) et opérationnel (court terme : quotidien à hebdomadaire). Ils sont caractérisés d'une part par l'horizon sur lequel portent les décisions qu'ils intègrent et d'autre part par les coûts qu'engendrent ces dernières. Chaque niveau définit un cadre pour celui qui lui est inférieur. Transversalement à ces degrés décisionnels, il existe une problématique majeure en gestion de production : la planification industrielle. Par suite, celle-ci se décompose également en trois niveaux hiérarchisés : les planifications stratégique, tactique et opérationnelle.

1. La planification stratégique (long terme) consiste, entre autres, en la définition de l'infrastructure du système de production (localisation d'usines, d'entrepôts...) et fournit un cadre physique à la planification tactique.

2. La planification tactique (moyen terme) s'intéresse à l'élaboration de plans de production c'est-à-dire détermine les quantités de produits à fabriquer par période afin de répondre au mieux à la demande, à un moindre coût.
3. La planification opérationnelle (court terme) consiste à ordonnancer la production définie au niveau tactique sur le système de production.

Dans cet article, nous nous intéressons à la problématique de la planification tactique. Depuis l'apparition de la méthode « Manufacturing Resources Planning » (MRPII) (Wight O., 1995), cette problématique est abordée de façon hiérarchisée en trois plans principaux : le Plan Industriel et Commercial (PIC), le Plan Directeur de Production (PDP) et le Calcul des Besoins Net (CBN ou MRP). Partant de la demande cliente prévisionnelle de produits et d'une estimation très grossière de la capacité de production, le PIC vise à l'obtention d'un compromis entre les objectifs de vente et de production. Les objectifs de production étant définis, l'élaboration du PDP consiste à planifier à l'aide de capacités estimées du système de production, la fabrication des produits finis. Or ceux-ci peuvent nécessiter un certain nombre de composants dont il faut également planifier la production : c'est le rôle du MRP. Dans la littérature, des modèles mathématiques dits de « lot sizing » ont été développés pour tenter de résoudre cette problématique. Ils permettent de répondre aux

problèmes d'élaboration des trois plans précédemment cités. D'une grande diversité, ils ont fait l'objet de plusieurs classifications (Drexl A. et Kimms A., 1997, Rizk N. et Martel A., 2001).

Dans cet article, nous nous focalisons sur l'élaboration du PDP. D'après (Drexl A. et Kimms A. 1997), il existe un modèle appelé « Capacitated Lot Sizing Problem » (CLSP) dédié à cette problématique. Ce dernier est connu comme étant NP-difficile (Bitran G.R. et Yanasse H.H., 1982). De fait, sa résolution fait l'objet d'une littérature importante notamment concernant l'utilisation de méthodes approchées. Cependant, comme le souligne (Brahimi N., 2004), l'emploi de métaheuristiques reste encore très marginale, et ceci malgré leur grande efficacité dans la résolution de problèmes NP-difficiles. L'objet de ce papier est donc de présenter une méthode d'optimisation efficace pour le CLSP, s'appuyant sur une métaheuristique à base de recuit-simulé afin d'élaborer des PDPs pour des instances de taille industrielle.

Notre article se décomposera de la façon suivante : dans une première partie, nous présenterons le modèle du CLSP, ces principales extensions ainsi qu'un rapide état de l'art des méthodes utilisées dans la littérature. La seconde partie sera consacrée à la description de la méthode d'optimisation proposée : plusieurs systèmes de voisinage seront présentés, une nouvelle fonction bi-objectif permettant d'une part de déterminer une solution réalisable du problème et d'autre part d'assurer l'accessibilité des solutions réalisables, sera donnée. Les premiers résultats numériques feront l'objet de la dernière partie et nous concluons par quelques perspectives.

2. LE CLSP

Dans cette partie, nous présentons successivement le modèle du CLSP, ses principales extensions et un bref état de l'art sur les méthodes de résolution rencontrées dans la littérature.

2.1 CLSP : modèle et quelques extensions

Le modèle du CLSP a été conçu dans le but d'élaborer un PDP. Ne s'intéressant donc qu'à la planification des produits finis, il fait partie de la famille des modèles de lot sizing « mono-niveau » contrairement aux modèles dits « multi-niveau » qui répondent également à la problématique du MRP. Il peut être considéré comme le modèle de référence parmi lesquels il s'inscrit (Comelli M. *et al.*, 2006).

Paramètres du CLSP :

- N : Nombre de type de produits à planifier
- T : Nombre de périodes constituant l'horizon de planification
- C_t : Capacité de production disponible durant la période t , $t \in \llbracket 1, T \rrbracket$
- d_{it} : Demande client en produit de type i durant la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$

- I_{i0} : Stock initial en produit i , $i \in \llbracket 1, N \rrbracket$
- p_i : Capacité requise pour produire un produit de type i , $i \in \llbracket 1, N \rrbracket$
- h_i : Coût de stockage pour un produit de type i durant une période, $i \in \llbracket 1, N \rrbracket$
- s_i : Coût de lancement de campagne pour le produit i durant une période, $i \in \llbracket 1, N \rrbracket$
- b_i : Coût de production d'un produit de type i , $i \in \llbracket 1, N \rrbracket$.

Variables de décision du CLSP :

- Q_{it} : Quantité de produit de type i fabriquée durant la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
- I_{it} : Quantité de produit de type i stockée en fin de la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
- X_{it} : Variable binaire indiquant si il y a eu un lancement de campagne pour le produit i durant la période t ($X_{it} = 1$) ou non ($X_{it} = 0$), $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$

En utilisant ces notations, le CLSP s'écrit :

$$\text{Minimiser } \sum_{i=1}^N \left[\sum_{t=1}^T (s_i X_{it} + b_i Q_{it} + h_i I_{it}) \right] \quad (1)$$

Sous les contraintes:

$$\begin{cases} I_{it} = I_{i(t-1)} + Q_{it} - d_{it} & \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket & (2) \\ p_i Q_{it} \leq C_t X_{it} & \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket & (3) \\ \sum_{i=1}^N p_i Q_{it} \leq C_t & \forall t \in \llbracket 1, T \rrbracket & (4) \\ X_{it} \in \{0,1\} & \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket & (5) \\ (Q_{it}, I_{it}) \in \mathbb{N}^2 & \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket & (6) \end{cases}$$

(1) est la fonction objectif à minimiser : elle calcule les coûts de production, de stockage et de lancement de campagne engendrés par le PDP déterminé, sur tout l'horizon de planification,

(2) est l'équation d'équilibre des stocks : le nombre de produit de type i en stock en fin de période t est égal au stock en fin de période $t - 1$ auquel s'ajoute la production en période t moins la demande client pour cette même période,

(3) représente la contrainte de lancement de campagne : en effet, dès que la moindre fabrication de produits de type i prendra place ($Q_{it} > 0$) alors nous aurons un lancement de campagne ($X_{it} = 1$)

(4) est la contrainte de capacité : par période, la capacité utilisée pour la production prévue ne peut excéder la capacité du système de production.

Les contraintes (5) et (6) sont les contraintes d'intégrité et de positivité.

Dans le modèle présenté, une solution admissible répondra nécessairement à la demande en temps voulu, laissant peu de flexibilité au système de production (pas de retard dans la production etc.). Pour remédier à ce

genre de problème, de nombreuses extensions ont été proposées, dont voici les principales :

- Backlogging (Zangwill W.I., 1989) : Cette extension permet de prendre en compte des retards de production. Pour ce faire, de nouvelles variables (I_{it}^-) mesurant le nombre de produits de type i en retard à la fin de la période t sont introduites. Bien évidemment, un coût de retard (h_i^-) est imputé dans la fonction objectif. Ainsi la fonction objectif (1) devient :

$$\text{Min.} \sum_{i=1}^N \left[\sum_{t=1}^T (s_i X_{it} + b_i Q_{it} + h_i I_{it} + h_i^- I_{it}^-) \right] \quad (7)$$

Et l'équation d'équilibre des stocks (2) :

$$I_{it}^- + I_{it} = I_{i(t-1)}^- + I_{i(t-1)} + Q_{it} - d_{it} \quad (8)$$

- Overtime (Diaby M. *et al.*, 1992) : plutôt que de prendre du retard dans la production, il peut être envisagé d'augmenter temporairement la capacité du système en ayant recours, par exemple, à des heures supplémentaires. Cette possibilité peut être modélisée en modifiant la contrainte de capacité (4) comme suit :

$$\sum_{i=1}^N p_i Q_{it} \leq C_t + O_t \quad (9)$$

où O_t est une variable représentant l'accroissement de la capacité pour la période t . De même que pour le backlogging, l'utilisation de ce surcroît de capacité est pénalisée par un coût unitaire (m_t) dans la fonction objectif qui devient alors :

$$\text{Min.} \sum_{t=1}^T \left[\sum_{i=1}^N (s_i X_{it} + b_i Q_{it} + h_i I_{it}) + m_t O_t \right] \quad (10)$$

- Setup Time (Trigeiro W. *et al.*, 1989) : dans certains cas, un lancement de campagne peut prendre un certain temps (il peut falloir reconfigurer les machines composant le système de production etc.). C'est pourquoi il peut être utile de le modéliser comme la consommation d'une certaine capacité du système de production. La contrainte de capacité (4) devient donc :

$$\sum_{i=1}^N (p_i Q_{it} + \tau_{it} X_{it}) \leq C_t \quad (11)$$

où τ_{it} représente la consommation de la capacité pour lancer la campagne pour le produit i durant la période t . (Trigeiro W. *et al.*, 1989) ont montré que l'utilisation de cette extension rendait le problème d'existence d'un PDP satisfaisant à ces contraintes, NP-Complet.

- A capacité infinie (Wagner H.M. et Whitin T.M., 1958) : Les capacités du système de production sont considérées comme infinies. (Wagner H.M. et Whitin T.M., 1958) ont montré qu'une solution optimale à ce problème vérifiait nécessairement la propriété du « zéro stock » qui dit qu'une période de production pour un produit de type i ne peut prendre place que si son stock est vide (cette propriété n'est pas vraie pour le CLSP à capacités quelconques). Un PDP vérifiant cette assertion est appelé « plan dominant ». En s'appuyant sur cette constatation, les auteurs ont développé un algorithme polynomial de programmation dynamique permettant de résoudre de manière optimale ce problème, rendant ainsi polynomial le CLSP en cas de capacité infinie.

2.2 CLSP : état de l'art

Dans ce bref état de l'art, nous nous concentrons sur les méthodes de résolution employées pour résoudre le CLSP dans sa version de base. Celles-ci peuvent être divisées en deux grandes familles : les méthodes exactes (programmation mathématique) qui visent à trouver la solution optimale du problème au prix, parfois, d'un temps de calcul prohibitif, et les méthodes approchées qui sacrifient la quête de l'optimum à la rapidité d'obtention d'une solution de bonne qualité.

Le CLSP étant NP-difficile (Bitran G.R. et Yanasse H.H., 1982), la littérature est surtout axée sur l'élaboration de méthodes approchées pour le résoudre : peu de méthodes exactes ont été développées. Bien que très efficaces, ces dernières trouvent leur limite dans les temps de calcul générés pour résoudre des instances de taille industrielle. Les différentes méthodes rencontrées peuvent être divisées en trois grandes familles :

- Les méthodes basées sur du Branch & Bound (Gelders *et al.*, 1986), (Hindi K.S., 1995)
- Les méthodes basées sur du Branch & Cut (Leung *et al.*, 1989), (Pochet Y. Wolsey L., 1991) etc.
- Les méthodes basées sur une reformulation du CLSP en problème de plus court chemin (Eppen G.D. et Martin R.K., 1987)

Concernant les méthodes approchées, on peut également rencontrer trois approches différentes :

- Les heuristiques de bon sens qui sont généralement des algorithmes gloutons. Parmi celles-ci on trouve des heuristiques qui construisent la solution période par période (Dixon P.S. et Silver E.A., 1981), produit par produit (Kirca Ö. et Kökten M., 1994) ou par amélioration successive (Dogramaci A. *et al.*, 1981).
- Les heuristiques à base de programmation mathématique qui peuvent utiliser des méthodes de relaxation (Thizy J.M. et Van Wassenhove L.N., 1985), des méthodes basées sur des techniques de programmation linéaire (en utilisant la reformulation en plus court chemin) (Gilbert K.C. et Madan M.S., 1991) ainsi que des méthodes de type Dantzig-Wolfe utilisant un modèle dit « approximatif » (Manne A.S.,

1958) qui ne s'intéresse qu'aux plans dominants (Huisman D. *et al.*, 2003).

- Les métaheuristiques. Peu de travaux ont été entrepris pour tenter de résoudre le CLSP en utilisant cette classe de méthodes (Brahimi N., 2004). Parmi ceux-ci, on peut citer les travaux de (Salomon M. *et al.*, 1993) qui utilisent un recuit simulé et une méthode tabou, (Haase K. et Kohlmorgen U., 1995) qui ont développé un algorithme génétique et (Hindi K.S., 1996) qui utilise une méthode tabou.

Les lecteurs intéressés par un état de l'art plus exhaustif pourront se référer à (Comelli M. *et al.*, 2006) ou à (Karimi B. *et al.* 2003).

3. METAHEURISTIQUES POUR LE CLSP

Le but de cet article est de présenter une métaheuristique à base de recuit simulé permettant d'obtenir des solutions de bonne qualité pour des instances de taille réelle (20 produits, 100 périodes). Ce type de métaheuristique utilise:

- un codage permettant de représenter les solutions du problème,
- un système de voisinage (une application permettant de passer d'une solution à une autre – son voisin –).

Une condition nécessaire au bon fonctionnement de ce type de méthodes est que toute solution soit accessible : si on considère deux solutions quelconques, on peut passer de l'une à l'autre par une ou plusieurs applications du système de voisinage. Nous proposons un système de voisinage qui, grâce à l'adjonction d'une nouvelle fonction bi-objectif permettant de s'extraire de l'espace des solutions admissibles pour le CLSP, vérifie cette hypothèse.

3.1 Définition du codage utilisé

Le CLSP utilise trois types de variables de décisions : les quantités fabriquées par produit et par période (Q_{it}), les quantités en stock en fin de période pour chaque produit (I_{it}) et les variables binaires qui indiquent si il y a eu ou non un lancement de campagne pour chaque produit durant la période considérée (X_{it}). Il s'ensuit que pour coder une solution du CLSP, il faudrait tenir compte de l'ensemble de ces variables. Cependant, l'équation des stocks (2) montre une correspondance biunivoque entre les variables de stocks et les variables de quantités. En effet, elle peut également s'écrire sous la forme:

$$Q_{it} = I_{it} - I_{i(t-1)} + d_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

ce qui prouve que les quantités fabriquées peuvent être déterminées de façon unique par les quantités stockées. Réciproquement, en sommant de façon adéquate les différentes équations des stocks, on montre que :

$$I_{it} = I_{i0} + \sum_{k=1}^t (Q_{ik} - d_{ik}) \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

Cela démontre qu'on peut également déterminer de façon unique les quantités stockées en fonction des quantités fabriquées. Cela montre que représenter les stocks dans notre codage serait redondant. En ce qui concerne les variables de lancement de campagne, il est clair qu'elles sont fonctions des quantités fabriquées. Plus précisément,

$$X_{it} = \chi_{]0, +\infty[}(Q_{it}) \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

où $\chi_{]0, +\infty[}$ est la fonction indicatrice de l'intervalle $]0, +\infty[$. Cette dernière n'étant pas bijective, nous pouvons conclure que connaissant les quantités produites il est possible d'en déduire de manière unique les lancements de campagne mais que la réciproque est fautive : en effet, connaissant les périodes où il y a un lancement de campagne, il n'est en général pas possible d'en déduire de manière unique les quantités fabriquées par produit et par période. (Salomon M. *et al.*, 1993) utilisent une matrice représentant les lancements de campagne et utilisent une heuristique gloutonne pour déterminer des quantités à fabriquer par produit et période qui soient les meilleures possibles. Cependant, ils ne peuvent assurer l'optimalité des quantités déterminées. Ceci montre qu'en restreignant de la sorte l'espace de recherche de leur recuit-simulé, ils ne peuvent assurer l'appartenance de la solution optimale à cette restriction.

Cette constatation nous amène donc à ne représenter que les variables de quantités. Ainsi, une solution sera représentée sous forme d'une matrice $Q \in \mathcal{M}_{N \times T}(\mathbb{N})$ où $\forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$, Q_{it} représente la quantité fabriquée en produit i durant la période t .

3.2 Le système de voisinage initial

Le système de voisinage doit permettre de passer d'une solution admissible à un ensemble de solutions admissibles. Pour le CLSP, une solution est admissible si et seulement si les quantités fabriquées permettent de répondre à la demande sans délai et si les capacités du système de production sont respectées. La première condition est exprimée par le fait que les variables de stocks sont toujours positives (contrainte (6)) et la seconde par la contrainte de capacité (4).

Soit $Q \in \mathcal{M}_{N \times T}(\mathbb{N})$ une solution admissible du CLSP. En partant de celle-ci, nous désirons déterminer une nouvelle solution admissible Q' . Pour cela nous allons devoir déplacer certaines quantités produites d'une période à une autre pour au moins un produit. Supposons que nous désirons déplacer une certaine quantité K de produits i fabriquée durant la période t à une autre période t' . Pour commencer, il est évident qu'on ne peut pas déplacer plus que la production de la période t . On a donc : $K \leq Q_{it}$. De plus, une fois cette quantité déplacée, on ne doit pas excéder la capacité de production à la période t' . La contrainte (4) nous dit donc que :

$$\sum_{j=1}^N p_j Q_{jt'} + p_i K \leq C_{t'}$$

p_i étant supposé non nul (sinon le problème devient polynomial), il vient :

$$K \leq \left\lfloor \frac{C_{t'} - \sum_{j=1}^N p_j Q_{jt'}}{p_i} \right\rfloor$$

En résumé nous devons donc avoir :

$$0 \leq K \leq \min \left(\left\lfloor \frac{C_{t'} - \sum_{j=1}^N p_j Q_{jt'}}{p_i} \right\rfloor, Q_{it} \right) \quad (12)$$

Cette condition nous assure le respect de la contrainte (4) mais pas celle de la positivité des stocks. En ce qui concerne les stocks, deux cas sont à considérer :

Cas 1 : $t' < t$

Comme l'illustre la figure 1, on préserve nécessairement la positivité des stocks (en effet, la solution Q satisfaisant à la demande, la solution Q' fera nécessairement de même puisqu'on ne fait que produire encore plus en avance).

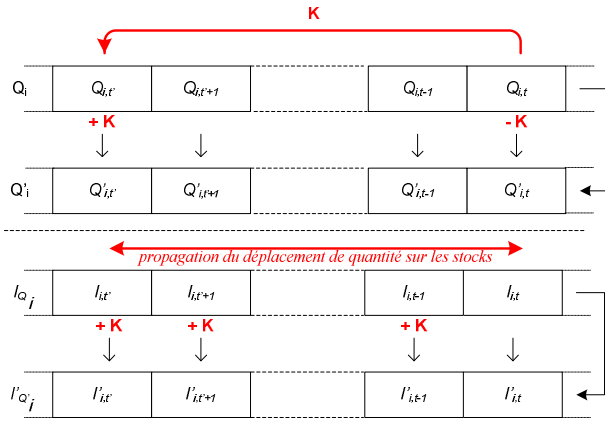


Figure 1. Effet du déplacement sur les stocks (cas 1)

Cas 2 : $t < t'$

La figure 2 nous montre comment garantir la positivité des stocks.

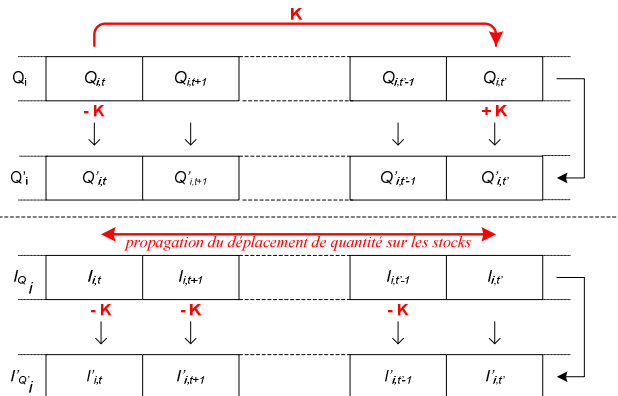


Figure 2. Effet du déplacement sur les stocks (cas 2)

En effet, déplacer la production de K produits de type i de la période t à période t' a pour effet de retrancher cette valeur aux stocks compris entre t et $t' - 1$. Donc,

pour préserver la positivité des stocks dans ce cas, il faut et il suffit que :

$$0 \leq K \leq \min_{t \leq l < t'} (I_{il}) \quad (13)$$

La conjonction de (12) et (13) permet d'énoncer la proposition suivante :

Proposition 1: Soit Q une solution admissible, pour qu'on obtienne une nouvelle solution admissible Q' en déplaçant une quantité de K produits de type i produite durant la période t à une période t' , il faut et il suffit que :

- Si $t < t'$:

$$0 \leq K \leq \min \left(\left\lfloor \frac{C_{t'} - \sum_{j=1}^N p_j Q_{jt'}}{p_i} \right\rfloor, Q_{it}, \min_{t \leq l < t'} (I_{il}) \right)$$

- Si $t' < t$:

$$0 \leq K \leq \min \left(\left\lfloor \frac{C_{t'} - \sum_{j=1}^N p_j Q_{jt'}}{p_i} \right\rfloor, Q_{it} \right)$$

Les différents voisinages que nous allons proposer vont s'appuyer sur cette proposition. Ceux-ci, au nombre de quatre, vont tour à tour jouer sur les principales sources de coûts : le stockage et le lancement de campagne.

3.2.1 Le voisinage 1 : déplacement non guidé

Le premier des voisinages proposés consiste à déplacer une quantité K respectant les conditions de la proposition 1. Le but de ce voisinage est double :

- éliminer un maximum de lancements de campagne,
- tenter de réduire les stocks.

Pour éliminer un lancement de campagne d'un produit i pour une période t , il suffit de déplacer la totalité de sa production sur une autre période t' . Pour favoriser cela, il suffit de prendre K le plus grand possible (K_{max}). Pour éliminer une quantité en stock, le choix systématique de K_{max} n'est pas judicieux dans le sens où les quantités à déplacer pour diminuer le stock peuvent être moindre. C'est pour cela qu'une quantité inférieure ou égale à K_{max} sera choisie.

K sera donc déterminé selon l'une des deux politiques que nous choisirons d'appliquer selon une certaine probabilité. Après plusieurs tests, nous avons choisi d'appliquer la politique K_{max} avec une probabilité de $\frac{2}{3}$. La figure 3 illustre le principe de fonctionnement du voisinage 1.

3.2.2 Le voisinage 2 : lancement

Le but de ce voisinage est de favoriser au maximum l'élimination des lancements de campagne les plus coûteux. Pour ce faire, les produits sont classés par ordre croissant de leur coût de lancement de campagne. Le choix du produit sur lequel le voisinage va agir est effectué aléatoirement. Cependant, on doit favoriser la désignation des plus coûteux : on va donc affecter à chaque produit, un intervalle dont la taille est fonction de son classement et qui permet ainsi de choisir prioritairement les plus coûteux. Ainsi, si le produit de

type i est à la $j^{\text{ème}}$ position du classement, on lui affectera l'intervalle $\llbracket 1 + \sum_{k=1}^{j-1} k, \sum_{k=1}^j k \rrbracket$ dont la taille est proportionnelle à son rang.

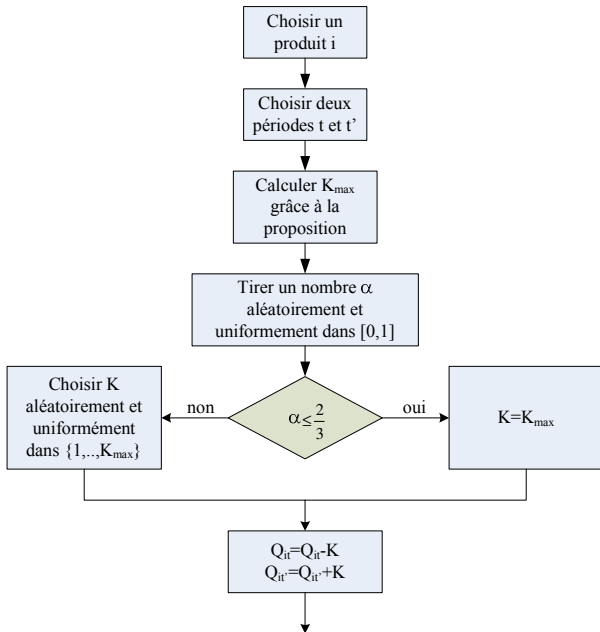


Figure 3. Algorithme de principe du voisinage 1

Le choix du produit sur lequel agit le voisinage s'effectue ainsi : un nombre α est tiré aléatoirement et uniformément dans l'intervalle $\llbracket 1, \frac{N(N+1)}{2} \rrbracket$ et appartiendra donc à l'intervalle d'un produit de type i . Une fois ce choix effectué, le voisinage procède de la façon suivante : on choisit deux périodes t et t' où il y a fabrication du produit i . Deux stratégies sont proposées :

- soit, à l'instar de la 1^{ère} politique du voisinage 1, on déplace la quantité maximale possible (K_{max}) de produit de type i de t à t' ,
- soit on essaie de déplacer l'intégralité de la fabrication de produit de type i de la période t , en la lissant sur des périodes où ce produit est également fabriqué. Suivant le choix de t' , on effectuera ce lissage de la manière suivante :

- si $t' < t$: pour p variant de $t - 1$ à 1, on regarde si nous avons de la fabrication de produit de type i durant la période p . Si oui, on calcule la quantité maximale déplaçable entre t et p : K_{max} et on déplace cette quantité de la période t à la période p . On s'arrête dès qu'on a pu éliminer toute la fabrication de produit de type i de la période t . Si on n'a pas pu déplacer l'intégralité de la fabrication à la période t , on laisse l'excédent de la production sur cette période.
- si $t < t'$, on lisse la production en partant de la période t' jusqu'à la première. On effectue les mêmes opérations que pour le cas $t' < t$, en faisant varier p de t' à 1, $p \neq t$.

Les deux stratégies seront choisies grâce à une probabilité : après plusieurs tests, nous avons choisi d'attribuer une probabilité de 0,8 pour la première

d'entre-elles. La figure 4 illustre le principe de fonctionnement du voisinage 2.

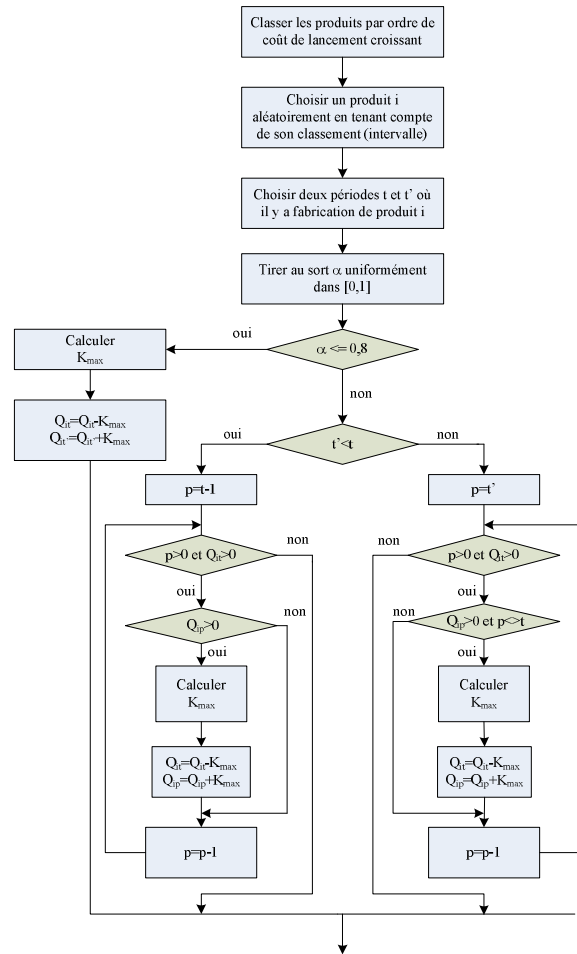


Figure 4. Algorithme de principe du voisinage 2

3.2.3 Le voisinage 3 : stock

Le but de ce voisinage est de favoriser au maximum l'élimination des stocks les plus coûteux. Pour ce faire, à l'instar du voisinage 2, on va classer les produits par ordre croissant des coûts de stockage et on procédera au choix du produit en utilisant la même technique que dans le voisinage précédent. Par ailleurs, son principe de fonctionnement est pratiquement identique à ceci près qu'on va choisir dans ce cas, deux périodes t et t' telles que $t < t'$ et, dans l'application de la seconde stratégie, nous ferons varier p de t' à t , en prenant garde de ne pas rajouter de coût de lancement. La figure 5 illustre le principe de fonctionnement du voisinage 3.

3.2.4 Le voisinage 4 : échange

Ce voisinage va permettre d'échanger la fabrication de deux types de produits i et j entre deux périodes t et t' . Pour commencer, on choisit aléatoirement les deux produits i et j et les deux périodes t et t' , en prenant $t < t'$. Ensuite, en supposant qu'on ne produit plus de produit de type j durant la période t' , on calcule grâce à la proposition 1 la quantité maximale de produit de type i qu'on pourrait déplacer de t à t' (K_{max}). Ensuite, on suppose déplacée cette quantité et on regarde si la

capacité du système à la période t' est dépassée et si oui, de combien. Dans ce cas, on tente de rétablir le respect de la capacité en déplaçant une partie (la plus petite possible) de fabrication du produit de type j de la période t' à la période t . Si cet objectif n'est pas atteint, on diminue la valeur de K_{max} d'une unité et on réitère le processus jusqu'à ce que l'on ait déterminé la plus grande quantité possible déplaçable de produit de type i de la période t à la période t' pouvant être compensée par un déplacement d'une partie de la fabrication du produit de type j de la période t' à la période t . Ensuite, il ne reste plus qu'à procéder aux déplacements de production. La figure 6 illustre le fonctionnement du voisinage 4.

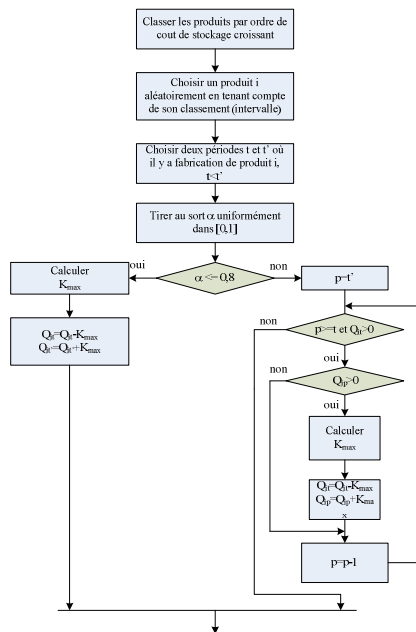


Figure 5. Algorithme de principe du voisinage 3

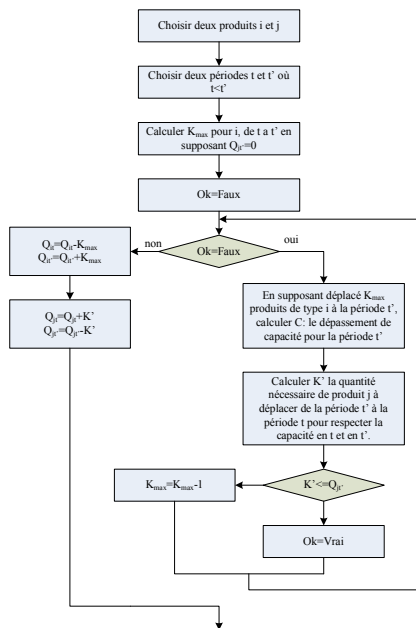


Figure 6. Algorithme de principe du voisinage 4

Le système de voisinage proposé initialement est composé des quatre voisinages précédents. Chacun d'entre eux est choisi aléatoirement de façon équiprobable.

3.3 La fonction bi-objectif

Pour le problème du CLSP, une solution est admissible si et seulement si la demande est satisfaite sans retard et si la capacité du système est respectée. Le système de voisinage proposé permet de passer d'une solution admissible à une autre : il se peut qu'il rencontre des problèmes dits d'accessibilité, c'est-à-dire qu'étant données deux solutions admissibles, il peut être très difficile, si ce n'est impossible, de passer de l'une à l'autre par une ou plusieurs applications du système de voisinage. Ce cas se rencontre surtout lorsque la capacité du système de production est à peine suffisante pour assurer la satisfaction de la demande.

Pour assurer l'accessibilité, il faudrait pouvoir s'affranchir des contraintes de capacité, c'est-à-dire permettre au système de voisinage de s'extraire de l'espace des solutions admissibles pour le CLSP. Dans la suite, nous qualifierons de solution toute solution non admissible répondant à la demande (autrement dit, qui ne respecte pas la capacité du système). L'ensemble des solutions admissibles pour le CLSP est donc un sous-ensemble de ces solutions.

Le but étant de trouver une solution admissible pour le CLSP, il faut pouvoir en favoriser l'obtention. Pour ce faire, à chaque solution Q , est associée une évaluation formée par un couple de valeurs $(Cap(Q), f(Q))$, où :

- $Cap(Q)$ est fonction du dépassement de capacité engendré par la solution Q ($Cap(Q)=0$ si les contraintes de capacité sont respectées, $Cap(Q)>0$ sinon),
- $f(Q)$ le coût de celle-ci (déterminé par la fonction objectif (1)).

On va favoriser l'obtention d'une solution admissible en appliquant une relation d'ordre totale sur l'ensemble des solutions de la manière suivante :

Soit deux solutions Q et Q' et soit $(Cap(Q), f(Q))$ et $(Cap(Q'), f(Q'))$ leurs évaluations respectives. On dira que Q est une meilleure solution que Q' si et seulement si :

- soit $Cap(Q) < Cap(Q')$
- soit $Cap(Q) = Cap(Q')$ et $f(Q) < f(Q')$.

Ainsi, toute solution admissible du CLSP sera meilleure que n'importe quelle solution non admissible pour le CLSP. Reste donc à définir la fonction $Cap(Q)$. Celle-ci doit vérifier la propriété suivante :

- $Cap(Q) = 0$ si la solution est admissible pour le CLSP
- $Cap(Q) > 0$ sinon.

Une solution est admissible pour le CLSP si et seulement si elle vérifie la contrainte (4) ce qui équivaut à

$$\sum_{t=1}^T \text{Max} \left(0, \sum_{i=1}^N p_i Q_{it} - C_t \right) = 0$$

La première idée était de considérer la fonction :

$$C_1(Q) = \sum_{t=1}^T \text{Max} \left(0, \sum_{i=1}^N p_i Q_{it} - C_t \right)$$

Cette idée s'apparente à l'extension « overtime » du CLSP où le coût de dépassement de capacité serait un majorant du coût de toute solution admissible du CLSP. Cependant, afin de trouver une solution admissible, il serait souhaitable que cette fonction permette de lisser la charge excédante. En effet, comme l'illustre la figure 7, la fonction $C_1(Q)$ peut donner la même valeur pour deux solutions qui ne sont pas équivalentes en termes de lissage de charge.

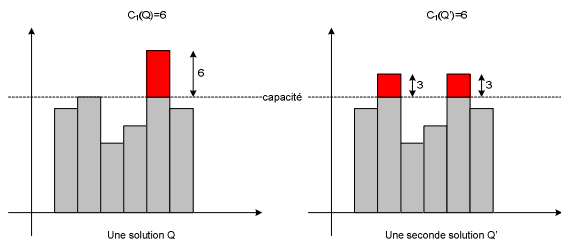


Figure 7. Deux solutions équivalentes pour la fonction C_1

Pour prendre en compte cet aspect, nous proposons la fonction $C_2(Q)$ définie par :

$$C_2(Q) = \sum_{t=1}^T \left[\text{Max} \left(0, \sum_{i=1}^N p_i Q_{it} - C_t \right) \right]^2$$

Cette fonction permet de favoriser le lissage de la charge comme le montre la figure 8 avec les deux solutions précédentes.

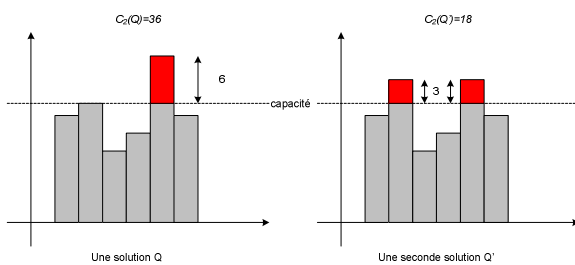


Figure 8. Evaluation des deux solutions par la fonction C_2

On choisit donc $Cap(Q) = C_2(Q)$.

3.4 Le système de voisinage définitif

Pour permettre de pleinement exploiter cette nouvelle fonction objectif et pour naviguer dans l'ensemble des solutions, deux nouveaux voisinages ont été rajoutés. Ceux-ci s'appuient sur la proposition 2, version de la proposition 1 ne tenant plus compte des contraintes de capacité :

Proposition 2: Soit Q une solution, pour qu'on obtienne une nouvelle solution Q' en déplaçant une quantité de K produit de type i produite durant la période t à une période t' , il faut et il suffit que :

- Si $t < t'$:
 $0 \leq K \leq \min \left(Q_{it}, \min_{t \leq l < t'} (I_{il}) \right)$
- Si $t' < t$:
 $0 \leq K \leq Q_{it}$

3.4.1 Le voisinage 5 : non guidé sans capacité

Ce voisinage reprend le voisinage 1 (déplacement non guidé) mais ne tient plus compte de la capacité de la période t' , c'est-à-dire que K_{max} est calculé grâce à la proposition 2.

3.4.2 Le voisinage 6 : échange sans capacité

Ce voisinage est une version simplifiée du voisinage 4 (échange). Ce voisinage va permettre d'échanger la fabrication de deux types de produits i et j entre deux périodes t et t' . Pour commencer, on choisit de façon aléatoire les deux produits i et j et les deux périodes t et t' , en prenant $t < t'$. On calcule grâce à la proposition 2 la quantité maximale de produit de type i qu'on pourrait déplacer de t à t' (K_{max}), ainsi que la quantité maximale de produit de type j qu'on pourrait déplacer de t' à t (K'_{max}). On déplace la quantité K de produit i choisie aléatoirement entre 1 et K_{max} et la quantité K' de produit j choisie aléatoirement entre 1 et K'_{max} . La figure 9 illustre le principe de fonctionnement du voisinage 6.

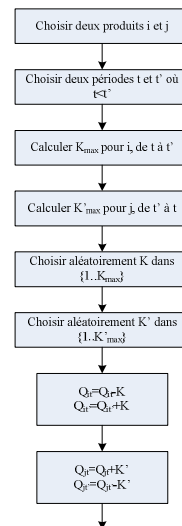


Figure 9. Algorithme de principe du voisinage 6

Le système de voisinage proposé est composé des six voisinages précédents. Chacun d'entre eux est choisi aléatoirement de façon équiprobable.

3.5 Schéma d'optimisation

Pour pouvoir juger de la qualité des solutions obtenues par les métaheuristiques, nous utilisons un calcul de

borne inférieure basée sur une relaxation lagrangienne des contraintes de capacité (4). (Chen W.H. et Thizy J.M., 1990) ont montré que c'est en relaxant ces contraintes qu'on obtient les meilleures bornes inférieures. Nous avons guidé cette relaxation grâce à une méthode de sous-gradient inspirée par celle de (Held M. Karp et R.M., 1970) qui a donné de très bons résultats pour le problème du voyageur de commerce. Cette relaxation lagrangienne permet d'une part d'obtenir une borne inférieure de bonne qualité et d'autre part une solution initiale pour notre méthode d'optimisation. En effet, seules les contraintes de capacités sont relaxées : Le plan de production donné par la relaxation lagrangienne respectera la demande client. Celui-ci sera donc bien une solution pour notre métaheuristique.

La figure 10 illustre le schéma de fonctionnement de notre méthode d'optimisation. Les voisinages sont symbolisés par la lettre V, leur choix étant équiprobable, une probabilité de $\frac{1}{6}$ leur est affectée.

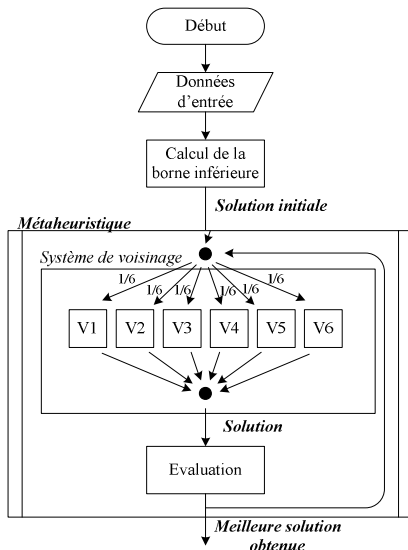


Figure 10. Schéma d'optimisation

4. RESULTATS

Dans cette partie, nous donnons les premiers résultats obtenus sur les 8 premières instances des 17 proposées (Eppen G.D. *et al.*, 1987). Le tableau 1 récapitule les résultats que nous avons obtenus. Dans celui-ci, nous donnons pour chaque instance :

- le nom et la taille (nombre de produits × nombre de périodes),
- [Best] le meilleur résultat connu (somme des coûts de stockage, lancement) donné par (Hindi K.S., 1996)
- [Tabou] le résultat obtenu par la méthode tabou présentée dans (Hindi K.S., 1996), qui est, à notre connaissance, la dernière proposition de méthode d'optimisation pour ce problème particulier.
- Le résultat obtenu par notre métaheuristique (la solution obtenue étant admissible, seule la composante coût est donnée). L'erreur relative entre

la solution obtenue et la meilleure solution connue ($Gap_{[b]}$)

$$Gap_{[b]} (\%) = \frac{Solution - [Best]}{[Best]} \times 100$$

entre la solution obtenue et la solution donnée par la méthode tabou (Hindi K.S., 1996) ($Gap_{[T]}$),

$$Gap_{[T]} (\%) = \frac{Solution - [Tabou]}{[Tabou]} \times 100$$

sont également mentionnées.

Les meilleurs résultats obtenus pour chaque instance ont été inscrits en gras.

Nom	Taille	[Best]	[Tabou]	Résultats obtenus		
				Obj.	Gap _[b]	Gap _[T]
E1	8×8	8430	8730	8430	0	-3,44
E2	8×8	7910	8250	7918	0,10	-4,02
E3	8×8	7610	7690	7610	0	-1,04
E4	8×8	7520	7520	7520	0	0
E5	50×8	48690	49001,9	48591	-0,20	-0,83
E6	100×8	94379,7	95681,3	94234	-0,15	-1,51
E7	150×8	137819	139755,2	137537	-0,20	-1,59
E8	200×8	188122	191274,4	187730	-0,20	-1,85

Tableau 1. Premiers résultats

Le tableau 2 donne, pour chaque instance, la borne inférieure obtenue ainsi que le Gap avec la solution obtenue.

Nom	Obj.	Résultats obtenus					
		Borne	Gap	Nom	Obj.	Borne	Gap
E1	8430	7837,59	7,5	E5	48591	48472,64	0,24
E2	7918	7722,27	2,53	E6	94234	93991,21	0,25
E3	7610	7564,16	1	E7	137537	137361,1	0,12
E4	7520	7464,16	0,74	E8	187730	187400,3	0,17

Tableau 2. Bornes inférieures

On voit que pour les instances de petite taille, on obtient la meilleure solution connue (à l'exception de E2 pour laquelle nous sommes à 0,1% de la meilleure solution connue) et que nous supplantons systématiquement la méthode tabou proposée par (Hindi K.S., 1996). On remarque également que pour les instances de grande taille, les résultats obtenus améliorent significativement la meilleure solution connue et que la borne inférieure que nous avons obtenue semble de bonne qualité. Nous poursuivons actuellement notre campagne de tests.

5. CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons proposé un nouveau système de voisinage ainsi qu'une nouvelle fonction bi-objectif pour la résolution du CLSP. Les premiers résultats obtenus montrent d'une part la pertinence de l'utilisation de métaheurstiques pour ce problème et d'autre part l'efficacité de notre méthode d'optimisation. Actuellement, nos tests se poursuivent sur les instances de (Eppen G.D. *et al.*, 1987). La fonction bi-objectif utilisée permet de guider la métaheuristique vers l'obtention d'une solution admissible pour le CLSP. Une perspective intéressante serait d'utiliser pour le problème de faisabilité rencontré avec l'extension « Setup Time ». Notre méthode d'optimisation pourrait donc être étendue à la résolution du CLSP « avec Setup

Time » (Le système de voisinage proposé pouvant s'adapter facilement pour prendre en compte des temps de lancement campagne).

REFERENCES

- Bitran G.R. and H.H. Yanasse, 1982. Computational complexity of the capacitated lot size problem, *Management Science*, 46(5), pp 724–738.
- Brahimi N., 2004. *Planification de la production : modèles et algorithmes pour les problèmes de dimensionnement de lots*. Thèse de doctorat, Université de Nantes, France.
- Chen W.H. and J.M. Thizy, 1990. Analysis of relaxation for the multi-item capacities lot-sizing problem. *Annal of Operations Research*, 26, p. 29–72.
- Comelli M., M. Gourgand and D. Lemoine, 2006. A review of tactical planning models, *International Conference Service System and Service Management - IEEE/ICSSSM06*, Troyes, France.
- Diaby, M., M.H. Bahl, H.C. Karwan and S. Zionts, 1992. A lagrangean relaxation approach for very-large-scale capacitated lot-sizing, *Management Science*, 38(9), p. 1329–1340.
- Dixon, P.S. and E.A. Silver, 1981. A heuristic solution procedure for the multi-item, single level, limited capacity, lotsizing problem, *Journal of operations management*, 2, p 23–39.
- Dogramaci A., J.C. Panayiotopoulos and N.R. Adam, 1981. The dynamic lot-sizing problem for multiple items under limited capacity. *AIIE Transactions*, 13(4), p. 294–303.
- Drexl A. and A. Kimms, 1997. Lot sizing and scheduling - Survey and extensions, *European Journal of Operational Research*, 99, p. 221-235.
- Eppen G.D. and R.K. Martin, 1987. Solving multi-item lot-sizing problems using variable redefinition. *Operations Research*, 35, p. 832-848.
- Gelders L.F., J. Maes and L.N. Van Wassenhove, 1986. A Branch and Bound Algorithm for the Multi-item Single Level Capacitated Dynamic Lot sizing Problem, *Multi-stage Production Planning and Inventory Control, Lectures Notes in Economics and Mathematical Systems*, 266, p.92–108.
- Giard V., 2003. *Gestion de la production et des flux*, 3^{ème} édition, Economica.
- Gilbert K.C. and M.S. Madan, 1991. A heuristic for a class of production planning and scheduling problems. *IIE Transactions*, 23, p 282–289.
- Haase K. and U. Kohlmorgen, 1995. Parallel genetic algorithm for the capacitated lotsizing problem, *Operations research proceedings*, p. 370–375.
- Held M. and R. M. Karp, 1970. The Traveling Salesman Problem and Minimum Spanning Trees, *Operations Research*, 18, p. 1138-1162.
- Hindi K.S., 1995. Computationally efficient solution of multi-item capacitated lot sizing problems, *Computers and industrial Engineering*, 28(4), p 709–719.
- Hindi K.S, 1996. Solving the clsp by a tabu search heuristic, *Journal of Operational Research Society*, 47(1), p. 151–161.
- Huisman, D., R. Jans, M. Peeters and A.P.M. Wagelmans, 2003. *Combining column generation and lagrangian relaxation*, Technical report.
- Karimi, B., S.M.T. Fatemi Ghomi and J. M. Wilson, 2003. The capacitated lot sizing problem: a review of models and algorithms, *Omega*, 31(5), p. 365-378.
- Kirca, Ö and M. Kökten, 1994. A new heuristic approach for the multi-item dynamic lot sizing problem. *European Journal of Operational Research*, 75, p. 332–341.
- Leung J.M., T.L. Magnanti and R. Vachani, 1989. Facets and algorithms for the capacitated lot sizing. *Mathematical programming*, 45, p 331–359.
- Manne A.S., 1958. Programming of economic lot sizes. *Management Science*, 4, p. 115–135,1958.
- Pochet, Y. and L. Wolsey, 1991. Solving multi-item lot-sizing problems using strong cutting planes, *Management Science*, 37, p. 53–67.
- Salomon M., R. Kuik and L.N. Van Wassenhove, 1993. Statistical search methods for lot-sizing problems. *Annals Operations Research*, 41, p. 453–468.
- Thizy J.M. and L.N. Van Wassenhove, 1985. Lagrangean relaxation for the multi-item capacitated lotsizing problem: a heuristic implementation, *IIE Transactions*, 17, p. 308–313.
- Wagner H.M. and T.M. Whitin, 1958. Dynamic version of the economic lot size model. *Management science*, 5, p.89–96.
- Wight O., 1995. *Manufacturing Resource Planning : MRP II*, John Wiley and Sons.
- Zangwill W.I., 1969. A backlogging model and a multi-echelon model of a dynamic economic lot size production system - a network approach. *Management Science*, 15, p.506-527.