Problème de livraisons à séquence fixée

Christophe Lenté, Yannick Kergosien

10ème Conférence Internationale de MOdélisation, Optimisation et SIMulation - MOSIM'14 - 5 au 7 Novembre 2014 - Nancy -France « de l'Économie linéaire à l'économie circulaire »

PROBLÈME DE LIVRAISONS À SÉQUENCE FIXÉE

Ch. LENTÉ, Y. KERGOSIEN

Université François Rabelais Tours
Laboratoire d'Informatique (EA 6300)
Equipe Ordonnancement et Conduite (ERL CNRS 6305)
64 avenue Jean Portalis - 37200 Tours - France
christophe.lente@univ-tours.fr, yannick.kergosien@univ-tours.fr

RÉSUMÉ : *Cet article présente plusieurs méthodes de découpage d'une séquence préfixée de clients à livrer par un unique véhicule. Lorsque l'ordre de livraison des clients pour un véhicule donné est imposé, le calcul des tournées de ce véhicule revient à déterminer le nombre de retours au dépôt et les dates correspondantes. Ce problème se complique rapidement dès que l'on prend en compte des contraintes telles que des dates de disponibilité des produits ou des dates de livraisons souhaitées. Les méthodes de résolution proposées varient suivant la nature du critère à optimiser (date de retour de la dernière tournée, nombre de livraisons en retard, ...). Les algorithmes présentés sont polynomiaux, inspirés de méthodes de programmation dynamique et destinés à être intégrés dans des méta-heuristiques afin de résoudre des problèmes plus complexes. Cette étude généralise également certains travaux de la littérature.*

MOTS-CLÉS : *Tournées de véhicules, livraisons, algorithme polynomial, programmation dynamique*

1 INTRODUCTION

L'étude présentée dans cette article tire son origine d'un travail réalisé pour l'Unité de Biopharmacie Clinique et Oncologique (UBCO) du CHRU de la ville Tours. Cette unité a pour mission de réaliser les préparations de chimiothérapies (médicaments anticancéreux) et de les livrer aux infirmières des services de soins pour l'administration aux patients. Cependant ces services étant éparpillés au sein d'un vaste complexe hospitalier, une personne est employée à plein temps à leurs livraisons. Après être produites, les préparations sont prises dans l'ordre de fin de production, afin d'être contrôlées par un automate. Le but de cette étape est de vérifier le bon dosage des différents composés de chaque préparation. Après l'analyse, les préparations sont stockées en attendant d'être prises en charge par le livreur. Lorsque le livreur rentre d'une tournée ou commence son travail, il a le choix entre (re)-partir immédiatement avec l'ensemble ou un sous-ensemble des préparations stockées ou attendre que d'autres préparations soient contrôlées afin de les emporter également. Cette décision est délicate. S'il attend trop longtemps, les premières préparations du stock peuvent être livrées en retard. Néanmoins, il est parfois préférable d'attendre la fin de l'analyse de certaines préparations, soit parce qu'elles risquent d'être livrées en retard à la prochaine tournée, soit pour être regroupées avec d'autres préparations à livrer dans un même service et éviter ainsi un ou plusieurs allers-retours inutiles. La problématique est donc de constituer des lots de produits faisant partie d'une même tournée.

Une métaheuristique a été développée, basée sur un séquencement des patients à livrer. Cette approche nécessite de résoudre de nombreuses fois un sous-problème de livraison à séquence fixée. Le problème consiste à trouver les dates de débuts de chaque tournée et les préparations qui composent chaque tournée, en fonction des dates de disponibilité des préparations (date de fin d'analyse), du nombre maximal de préparations que peut transporter le livreur et des dates de livraison souhaitées par les services.

Cette approche est assez répandue pour résoudre des problèmes de tournées de véhicule (*VRP - Vehicle Routing Problem*) : elle consiste à construire une séquence de clients à livrer puis à évaluer cette séquence. L'évaluation consiste à découper cette séquence et à affecter un ou des véhicules à chacune des sous-séquences de manière à obtenir une solution réalisable. C'est une approche particulièrement bien adaptée à des métaheuristiques. On peut citer par exemple les traveaux présentés dans (Prins C., 2004) pour résoudre le VRP ou ceux de (Bouly H. et al., 2010) où les auteurs présentent une méthode pour décomposer un tour géant afin de maximiser des profits. Cependant, ces dernières études supposent qu'un véhicule n'effectue qu'une unique tournée, à l'inverse

du cas qui nous intéresse où plusieurs tournées sont envisagées pour un unique véhicule. La résolution d'un problème de tournées de véhicules avec plusieurs tournées par véhicule (*Multi-Trip Vehicule Routing Problem*) a déjà fait l'objet de plusieurs études telles que celles présentées dans (Olivera A. et al., 2007), (Macedo R., 2011) et (Cattaruzza D. et al., 2013). Dans (Cattaruzza D. et al., 2013), les auteurs ont notamment adapté la méthode de découpage de (Prins C., 2004). Cette adapation passe par l'exécution d'un algorithme d'étiquette dans un graphe pour l'affectation des tournées aux véhicules.

Ces algorithmes de découpage de tournées s'avèrent également utiles pour la résolution de problèmes d'ordonnancement couplés à des problèmes de livraison (Chen Z.L. et G.L. Vairaktarakis, 2005). Dans ce type de problème, plusieurs tâches doivent être ordonnancées sur une ou plusieurs machines puis transportées vers une ou plusieurs destinations par un ou plusieurs livreurs. Ce type de problème n'a que récemment été étudié dans la littérature et offre de nombreuses applications, nous pouvons notamment citer (Toptal A. et al., 2014), (Lee J. et al., 2014) et (Chen H-K. et al.,2009). Certaines études se sont intéressées à des cas simples, de complexité polynomiale, ou à développer des programmes dynamiques pour d'autres cas plus complexes, comme (Li C-L. et al., 2008). Dans cette dernière étude, les auteurs s'intéressent à un problème à une machine et un livreur qui effectue plusieurs tournées. Leur objectif est de minimiser la somme des dates de livraison.

Le problème étudié dans cet article est issu d'une étude plus globale sur la production et la livraison de chimiothérapies (Kergosien Y. et al., 2011), il constitue également une large généralisation de celui abordé dans (Tsirimpas P. et al., 2008). Dans cette dernière étude, les auteurs abordent un problème mono-véhicule avec une séquence de livraison fixée, le problème consiste à planifier les retours au dépôt du véhicule.

2 PROBLÈME ET NOTATIONS

Une liste des notations utilisées est donnée en appendice à la fin de l'article.

Dans ce problème, un ensemble de n clients doivent être approvisionnés en K types de produits différents par un unique véhicule. Les clients sont numérotés de 1 à n et doivent être livrés dans cet ordre. Chaque client i ($1 \leq i \leq n$) a des demandes q_{ik} en produit k ($1 \leq k \leq K$). Ces produits ne sont pas nécéssairement disponibles au dépôt pour être livrés dès le départ de la première tournée, par exemple parce qu'ils doivent être d'abord confectionnés. Ainsi des dates de disponibilités au plus tôt des produits au dépôt doivent être prises en compte. Les produits k destinés au client i ne sont pas disponibles avant la date r_{ik}, par contre tous les produits destinés à un même client doivent lui être livrés en même temps et ceci avant une date souhaitée de livraison d_i. Les durées de déplacement δ_{0i} du véhicule entre le dépôt et un client i ainsi que les durées de déplacements $\delta_{i(i+1)}$ entre deux clients consécutifs, sont connues. Le véhicule peut avoir soit un unique compartiment de capacité Q, dans lequel tous les produits sont mélangés, soit K compartiments de capacités Q_k ($1 \leq k \leq K$), un par type de produit, ou tout autre aménagement intermédiaire. Pour les indications de complexité des diverses algorithmes présentés, on notera, suivant le véhicule envisagé, $\tilde{Q} = Q$ ou $\tilde{Q} = \sum_{k=1}^{K} Q_k$.

On peut remarquer qu'une tournée devant livrer des clients, jusqu'au client j, quittera au plus tôt le dépôt à la date $r_j = max_{1 \leq k \leq K} r_{jk}$, puisqu'il faut livrer tous les produits d'un client en une seule fois. D'autre part, on peut supposer sans perte de généralité que les dates r_j sont numérotées par ordre croissant puisque qu'un client j sera toujours livré avant un client $j+1$. C'est l'hypothèse que l'on posera pour la suite. La taille des tournées est limitée par la ou les capacité(s) du véhicule. Pour chaque client j, on peut calculer n_j, le nombre maximum de clients pouvant être servis avant lui dans une même tournée, j compris. Le calcul de l'ensemble des valeurs n_j peut s'effectuer en $O(n)$.

On souhaite minimiser séparément plusieurs critère :

- La dernière date de retour au dépôt (C_{max}),
- La distance parcourue,
- Le plus grand retard algébrique de livraison (L_{max}),
- Le nombre de clients livrés en retard ($\sum U_i$).

Résoudre ces problèmes consiste essentiellement à déterminer quels groupes de clients vont être livrés dans une même tournée ou, de manière équivalente, après quels clients le véhicule doit revenir au dépôt pour s'approvisionner. Imposer l'ordre des livraisons simplifie le problème mais pas suffisamment pour rendre sa résolution évidente. Par contre cela permet de rendre le problème polynomial. Les méthodes employées s'appuient sur la mise en évidence d'un graphe sous-jacent au problème (cf. figure 1) et sur la propagation d'étiquettes sur les sommets de ce graphe, suivant un principe similaire à l'algorithme de Dijkstra.

Dans ce graphe, les sommets représentent des retours aux dépôts et les arcs des tournées. Plus précisément, le sommet D_i symbolise le retour du véhicule

au dépôt juste après avoir servi le client i, D_0 symbolisant le premier départ. L'arc (D_i, D_j) représente la tournée de livraison des clients $i+1$ à j, cet arc n'a évidemment de sens que si i appartient à l'intervalle $[\![j - n_j, j - 1]\!]$. La notation $[\![.,.]\!]$ est utilisée pour indiquer un intervalle dans l'ensemble des entiers. Une solution au problème, en conséquence, est un chemin de D_0 à D_n. Par exemple, sur la figure 1, le chemin en gras correspond à la solution composée d'une première tournée qui livre les clients 1 à i puis d'une deuxième tournée livrant les clients $i + 1$ à j suivie d'une troisième tournée passant par les clients $j+1$ à n. Entre chaque tournée le véhicule revient se réapprovisionner au dépôt. Le graphe possède donc $n+1$ sommets et au plus $n.\tilde{Q}$ arcs qui représentent toutes les tournées possibles en tenant compte de la capacité du véhicule (le pire des cas correspondant à une unique unité de produit demandée par chaque client).

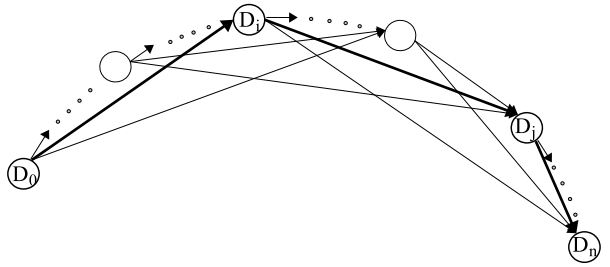

Figure 1: Graphe sous-jacent

3 MINIMISATION DE LA DERNIÈRE DATE DE RETOUR AU DÉPÔT

La méthode de résolution consiste à calculer par propagation des dates d'arrivée au plus tôt à chaque sommet D_i. Pour cela, on définit pour tout sommet D_j une date C_j qui est la date au plus tôt de retour au dépôt après avoir livré tous les clients de 1 à j ainsi qu'un ensemble de dates $C(j|i)$ qui sont les dates au plus tôt de retour au dépôt après avoir livré tous les clients jusqu'à j, si la dernière tournée visite les clients $i+1$ à j. Enfin, on note $\Delta_{i,j}$ la durée nécessaire, dépôt à dépôt, pour livrer les clients $i+1$ à j, en une unique tournée.

Les durées $\Delta_{i,j}$ peuvent être précalculées simplement par la formule :

$$\forall j \in [\![1, n]\!],\ \forall i \in [\![j - n_j, j - 1]\!],$$
$$\Delta_{i,j} = \delta_{0(i+1)} + \sum_{k=i+1}^{j-1} \delta_{k(k+1)} + \delta_{0j} \quad (1)$$

En posant $C_0 = 0$, les deux formules suivantes unissent les dates C_j et $C(j|i)$:

$$\forall j \in [\![1, n]\!],\ \forall i \in [\![j - n_j, j - 1]\!],$$
$$C(j|i) = max(C_i, r_j) + \Delta_{ij} \quad (2)$$

et

$$\forall j \in [\![1, n]\!],\ C_j = \min_{i \in [\![j-n_j, j-1]\!]} C(j|i) \quad (3)$$

La valeur de C_n correspond à la plus petite date de retour au dépôt après livraison du dernier client. Elle peut être calculée en $O(n.\tilde{Q})$.

Les formules précédentes se basent sur la propriété de monotonie (4). Supposons que s_1 et s_2 soient des solutions pour un sous-problème limité aux i premiers clients et s_3 un ensemble de tournées sur les $n - i$ derniers clients alors ($s_u s_v$ représentant la mise en séquence des deux ensembles de tournées) :

$$C_{max}(s_1) \leq C_{max}(s_2) \Rightarrow C_{max}(s_1 s_3) \leq C_{max}(s_2 s_3) \quad (4)$$

4 MINIMISATION DE LA DISTANCE PARCOURUE

La méthode précédente peut s'appliquer à la minimisation de la distance parcourue par le véhicule. Il suffit de poser toutes les dates de disponibilités r_j à zéro et de renseigner les variables $\delta_{i(i+1)}$ non pas par des durées mais par des distances.

5 MINIMISATION DU PLUS GRAND RETARD ALGÉBRIQUE

La méthode pour calculer les dates de retour s'appuie sur le fait qu'une solution optimale pour n clients se construit en prolongeant une solution optimale pour m clients ($m < n$). Cette propriété devient fausse dans le cas du plus grand retard algébrique : privilégier une solution engrendrant de petits retards sur les premiers clients ne mènera pas forcément à une solution optimale. La figure 2 illustre cette difficulté dans un cas mono-produit et avec un véhicule de capacité supposée infinie.

Dans cet exemple, la solution optimale consiste à livrer les deux premiers clients dans une première tournée puis le client 3 dans une tournée séparée. Cela mène à un retard maximal de 4. Par contre, si on ne s'intéresse qu'aux deux premiers clients, il vaut mieux les livrer dans deux tournées séparées que dans une unique tournée. Ainsi on obtiendrait un retard maximal de -1 pour la livraison de ces deux premiers clients, contre un retard de 4 s'ils avaient été livré ensemble. Le client 3 serait alors livré soit seul dans une troisème tournée, soit avec le deuxième client, ce qui mène dans les deux cas à un retard de 5. Cette solution ne serait donc pas optimale.

La propriété 4 ne s'étend donc pas au retard al-

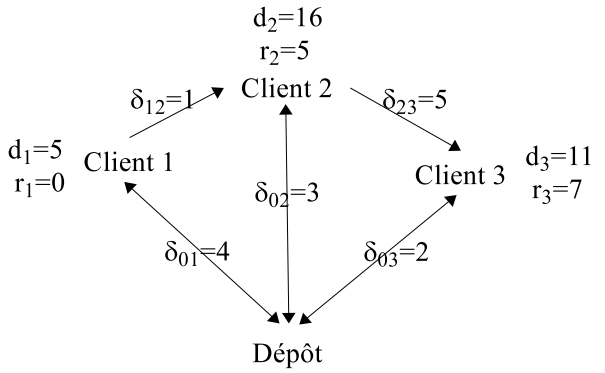

Figure 2: Exemple de calcul des retards algébriques

gébrique maximal :

$$L_{max}(s_1) \leq L_{max}(s_2) \not\Rightarrow L_{max}(s_1 s_3) \leq L_{max}(s_2 s_3) \quad (5)$$

En notant $L_{max}(s)$ le plus grand retard de livraison observé durant les tournées de s.

Il subsiste néanmoins une autre relation de dominance, moins forte :

$$\left.\begin{array}{l} C_{max}(s_1) \leq C_{max}(s_2) \\ L_{max}(s_1) \leq L_{max}(s_3) \end{array}\right\} \Rightarrow L_{max}(s_1 s_3) \leq L_{max}(s_2 s_3) \quad (6)$$

Cette relation de dominance suggère, pour contourner la difficulté, d'adopter une approche multi-critère et plus précisément ici, une approche $\epsilon-$contrainte. Pour cela, il faut définir de nouvelles variables. On note C_j^ϵ la date au plus tôt de retour au dépôt après avoir livré tous les clients de 1 à j avec un retard maximal inférieur à ϵ, $C^\epsilon(j|i)$ la date au plus tôt de retour au dépôt après avoir livré tous les clients jusqu'à j avec un retard maximal inférieur à ϵ, si la dernière tournée visite les clients $i+1$ à j. On note ensuite $L_{max}^\epsilon(j|i)$ le plus petit retard de livraison maximal possible des clients $i+1$ à j s'ils sont tous livrés dans la même tournée et que les clients précédents (de 1 à i) ont tous été livrés avec un retard maximal de ϵ. Enfin on pose, pour chaque arc, $MinL_{max}(i,j) = \max\limits_{i+1 \leq k \leq j} \left(\delta_{0(i+1)} + \sum_{\ell=i+1}^{k} \delta_{\ell(\ell+1)} - d_k\right)$ la valeur du retard de livraison maximal des clients $i+1$ à j s'ils sont tous livrés dans une même tournée qui part à la date zéro, ce qui correspond également au plus petit retard maximal envisageable.

Ces diverses variables sont reliées par les trois formules suivantes :

$$L_{max}^\epsilon(j|i) = max(r_j, C_i^\epsilon) + MinL_{max}(i,j) \quad (7)$$

$$\forall j \in [\![1,n]\!],\ \forall i \in [\![j-n_j, j-1]\!],\ tq\ L_{max}^\epsilon(j|i) \leq \epsilon : \quad C^\epsilon(j|i) = max(C_i^\epsilon, r_j) + \Delta_{ij} \quad (8)$$

et

$$C_j^\epsilon = \min_{\substack{i \in [\![j-n_j, j-1]\!] \\ L_{max}^\epsilon(j|i) \leq \epsilon}} C^\epsilon(j|i) \quad (9)$$

En adoptant comme convention que $min(\emptyset) = +\infty$, il est possible de vérifier en $O(n\tilde{Q})$ s'il existe une solution avec un retard algébrique maximum de livraison inférieur ou égal à ϵ : il suffit de vérifier si C_n^ϵ est fini ou non. Dans le cas où il n'existe pas de solution, les calculs peuvent s'arrêter au premier C_j^ϵ infini rencontré. Ainsi, en augmentant progressivement ϵ jusqu'à obtenir une solution réalisable, la solution optimale pourra ête obtenue. La méthode est présentée dans son intégralité dans l'algorithme 1, dont la complexité est au pire en $O(n^2\tilde{Q}^2)$. En effet, dans le pire des cas, ϵ est réévalué $n\tilde{Q}$ fois puisqu'à chaque réévaluation un nouveau $C^\epsilon(j|i)$ est défini.

Algorithme 1 : Minimisation du plus grand retard algébrique.

```
1  ε ← min_{j∈[1,n]} min_{i∈[j−n_j, j−1]} r_j + MinL_max(i,j)
2  C_0^ε ← 0
3  pour j de 1 à n faire
4      C_j^ε ← +∞
5  Prochain_ε ← +∞
6  j ← 1
7  tant que (j < n) faire
8      pour i de j − n_j à j − 1 faire
9          Calculer L_max^ε(j|i)
10         si L_max^ε(j|i) ≤ ε alors
11             C_j^ε = min(C_j^ε, max(C_i^ε, r_j) + Δ_ij)
12         sinon
13             Prochain_ε ←
               min(Prochain_ε, L_max^ε(j|i))
14     si C_j^ε = +∞ alors
15         ε ← Prochain_ε
16         Prochain_ε ← +∞
17         j ← 1
18     sinon
19         j ← j + 1
20 retourner ε
```

6 MINIMISATION DU NOMBRE DE TRAVAUX EN RETARD

La minimisation du nombre de travaux en retard se heurte aux mêmes difficultés que celle du retard algébrique maximal : une solution partielle livrant très

peu de clients en retard parmi les premiers clients ne conduira pas nécessairement à une solution optimale. Il faut donc de nouveau envisager une approche multicritère, mais cette fois-ci le critère du nombre de clients livrés en retard admet au maximum n+1 valeurs distinctes. On peut donc calculer un front de Pareto, de taille au plus $n+1$, et le propager de sommets en sommets dans le graphe sous-jacent. Ce front portera sur le couple de critères (Date d'arrivée au plus tôt au sommet D_j, Nombre de clients livrés en retard parmi les j premiers clients).

On note $C_j(k)$ la plus petite date d'arrivée possible au sommet D_j si exactement k clients ont été livrés en retard parmi les j premiers. On note également $NbR(i,j,k)$ le nombre minimum de clients en retard dans la tournée livrant les clients de $i+1$ à j, si exactement k clients ont été livrés en retard parmi les i premiers. Ces variables sont liées par la formule suivante :

$$\begin{array}{l} \forall j \in [\![1,n]\!],\ \forall k \in [\![0,j]\!] : \\ C_j(k) = \\ \qquad \min \quad (max(r_j, C_i(k')) + \Delta_{ij}) \\ \quad i \in [\![j-n_j, j-1]\!] \\ \quad k' + NbR(i,j,k') = k \end{array} \quad (10)$$

L'algorithme 2 permet de calculer l'ensemble des solutions qui constituent le front de Pareto pour les critères (dernière date de retour au dépôt, nombre de clients livrés en retard). Cet algorithme est de complexité $O(n^2\tilde{Q})$.

Algorithme 2 : Minimisation du nombre de livraisons en retard.

```
pour j de 1 à n faire
    pour k de 0 à j faire
        C_j(k) ← +∞
pour j de 1 à n faire
    pour i de j − n_j à j faire
        pour k' de 0 à i faire
            k ← k' + NbR(i, j, k')
            C_j(k) ←
            min(C_j(k), max(r_j, C_i(k')) + Δ(i, j))
retourner C_n(k)
```

7 RÉSULTATS

7.1 Amélioration d'algorithmes existants dans la littérature

Dans (Tsirimpas P. et al., 2008), les auteurs arbordent une problématique semblable. Ils ont proposé plusieurs méthodes exactes, basées sur la programmation dynamique, afin de résoudre trois variantes de problèmes de livraison à séquence fixée avec un seul livreur. La première variante concerne le cas où plusieurs types de produits doivent être livrés, sachant que chaque produit est transporté dans des compartiments distincts. La deuxième variante est identique à la précédente, hormis le fait que les produits sont transportés dans un même compartiment. Et la dernière variante suppose qu'il y a un unique type de produit à livrer et à collecter. Les auteurs ne prennent pas en considération de dates de disponibilités ni de dates de livraisons souhaitées. Ils ne s'intéressent qu'au seul critère de la distance totale parcourue. Ces trois problèmes sont donc un cas particulier du problème présenté en section 4 : toutes les dates de disponibilités r_i sont égales à zéro et donc le critère de la distance totale parcourue est équivalent au critère de la date de retour après la dernière livraison. Quelque soit la variante des problèmes proposés par les auteurs, le même algorithme présenté en section 3 s'applique puisque la prise en compte des différences entre ces variantes s'effectue dans la construction du graphe sous jacent (cf. figure 1). En effet, elle s'effectue dans la définition des arcs du graphe, plus exactement dans le calcul des n_j. Dans le cas d'un véhicule à compartiment unique, l'algorithme qu'ils proposent et celui présenté ici ont la même complexité $O(nQ)$. Par contre dans le cas d'un véhicule multi compartiment, ils proposent un algorithme en $O(n.\prod Q_k)$ qui croît donc exponentiellement avec le nombre de compartiments contrairement à l'algorithme en $O(n.\sum Q_k)$ présenté ici.

7.2 Expérimentation numérique

Afin de tester les temps de résolution des algorithmes proposés, nous avons généré 18 ensembles de 100 instances. Chaque ensemble d'instance est caractérisé par un nombre de clients égal à 50, 100, 200, 500, 700 ou 1000, et par la capacité du véhicule égale à 10, 15 ou 20 fois la quantité moyenne demandée par l'ensemble des clients d'une même instance. Ainsi, le véhicule pourra livrer en moyenne k_c=10, 15 ou 20 clients maximum dans une même tournée. Un seul type de produit est demandé. Pour chaque client la quantité demandée en produit est tirée aléatoirement entre 1 et 10. Les coordonnées des clients sont générées dans un carré de côté 50 et la distance Euclidienne est utilisée pour calculer les durées. Les dates souhaitées de livraison sont générées aléatoirement de manière à obtenir des écarts compris entre 10 et 50 entre chaque d_{i-1} et d_i (d_1 étant tiré aléatoirement entre 10 et 50). Les dates de disponibilités r_i des produits sont générées de la même manière, cependant les $k_c/2$ premières dates sont nulles. Les algorithmes ont été implémentés en langace C et les tests ont été effectués sur processeur AMD Phenom II X6 1090T,

3.2 GHz et 8 Go RAM.

Le tableau 1 présente les temps moyens de résolution des algorithmes obtenus pour chaque type (n,k_c) d'ensemble d'instances. La ligne C_{max} concerne la minimisation de la dernière date de retour au dépôt, L_{max} la minimisation du plus grand retard algébrique et $\sum U_i$ la minimisation du nombre de livraisons en retard.

D'un point de vue pratique, jusqu'à 200 clients, la résolution des problèmes est instantanée. Dans tous les cas, sauf le dernier, les temps de résolution restent inférieurs au dixième de seconde. Il faut noter que la capacité du véhicule influe grandement sur le temps de résolution. Une capacité infinie amène l'algorithme de minimisation du L_{max} à une complexité de $O(n^4)$ et celui de minimisation du nombre de retards à une complexité de $O(n^3)$. Mais en pratique, le critère de minimisation du nombre de retards correspond aux problèmes qui demandent le plus de temps de résolution, puisqu'un front de Pareto entre ce critère et la date de retour de la dernière tournée est calculé, tandis que pour le L_{max} le nombre maximal de répétitions n'est jamais atteint.

CONCLUSION ET PERSPECTIVES

Dans cet article, nous nous sommes intéressés à un problème de livraisons à séquence fixées généralisant une étude déjà présente dans la littérature. Trois différents critères à minimiser ont été étudiés : la date de retour au dépôt après la dernière tournée, le plus grand retard de livraison et le nombre de retards de livraison. Dans chaque cas un algorithme de résolution exact et polynomial a été proposé, améliorant les méthodes déjà existantes. Jusqu'à 1000 clients, ces méthodes sont intégrables sans problème dans une métaheuristique.

Une généralisation immédiate consisterait à résoudre un problème multicritère sur le nombre de livraisons en retard et le retard algébrique maximal de livraison. La somme des retards absolus pourrait également être étudié. Enfin, afin d'aborder des problèmes de tournées de véhicules avec fenêtre de temps, il serait intéressant d'adapter ces algorithmes pour la prise en compte des dates de livraison au plus tôt chez le client. Celles-ci influenceraient uniquement le calcul des Δ_{ij} qui dépendront également de la date de départ du véhicule pour livrer les clients $i+1$ à j. Cependant les complexités des algorithmes pour la minimisation du C_{max} et du L_{max} augmenteraient d'un facteur Q, celle pour $\sum U_i$ resterait inchangée.

APPENDICE

Liste des notations utilisées.

- n clients numérotés de 1 à n sont à livrer dans cet ordre,
- K produits distincts,
- le client n° i souhaite une quantité q_{ik} du produit k ($1 \leq k \leq K$),
- il y a un unique véhicule de livraison.
  - de capacité Q (un seul compartiment)
  - de capacités Q_k (un compartiment par type de produit)
- le produit k destiné au client i n'est disponible qu'à partir de la date r_{ik},
- $r_i = max_{1 \leq k \leq K} r_{ik}$, la date avant laquelle on ne peut pas lancer une tournée pour livrer le client i,
- le client n° i souhaite être livré avant la date d_i et tous les produits doivent lui être livrés en une seule fois,
- n_i le nombre maximum de clients pouvant être servis dans une même tournée avant le client i, i compris,
- $\delta_{j,j+1}$ le temps de trajet entre les clients i et $i+1$,
- $\delta_{0,i}$ le temps de trajet entre le dépôt et le client i,
- $\Delta_{i,j}$ la durée nécessaire, dépôt à dépôt, pour livrer les clients $i+1$ à j, en une unique tournée.
- C_j la date au plus tôt de retour au dépôt après avoir livré tous les clients de 1 à j,
- $C(j|i)$ la date au plus tôt de retour au dépôt après avoir livré tous les clients jusqu'à j, si la dernière tournée visite les clients $i+1$ à j.
- C_j^ϵ la date au plus tôt de retour au dépôt après avoir livré tous les clients de 1 à j avec un retard maximal inférieur à ϵ,
- $C^\epsilon(j|i)$ la date au plus tôt de retour au dépôt après avoir livré tous les clients jusqu'à j avec un retard maximal inférieur à ϵ, si la dernière tournée visite les clients $i+1$ à j.
- $L_{max}^\epsilon(j|i)$ le retard de livraison maximal minimal des clients $i+1$ à j s'ils sont tous livrés dans la même tournée et que les clients précédents (de 1 à i) ont tous été livrés avec un retard maximal de ϵ.
- $MinL_{max}(i,j) = \underset{i+1 \leq k \leq j}{max} \left(\delta_{0(i+1)} + \sum_{\ell=i+1}^{k} \delta_{\ell(\ell+1)} - d_k \right)$ la plus petite valeur du retard de livraison maximal des clients $i+1$ à j s'ils sont tous livrés dans une même tournée qui part à la date zéro..

n	50			100			200		
k_c	10	15	20	10	15	20	10	15	20
C_{max}	0,00342	0,00426	0,00462	0,00652	0,00792	0,00930	0,013250	0,01727	0,01911
L_{max}	0,09148	0,09993	0,07491	0,32450	0,46495	0,24351	1,25449	1,38641	1,25218
$\sum U_i$	0,09208	0,19057	0,24373	0,34056	0,67091	1,24080	1,46645	3,26349	5,49812

n	500			700			1000		
k_c	10	15	20	10	15	20	10	15	20
C_{max}	0,03032	0,03917	0,04531	0,04257	0,05570	0,06538	0,06141	0,07844	0,09124
L_{max}	4,39991	5,25450	6,13320	9,73071	10,6313	11,3984	14,4746	17,3922	21,6666
$\sum U_i$	7,62170	19,5554	33,4501	13,4811	38,4715	72,2463	30,7718	70,5179	119,151

Tableau 1: Temps de résolution (en ms)

Les critères :

- C_{max} : la dernière date de retour au dépôt,
- la distance parcourue,
- L_{max} : le plus grand retard algébrique de livraison,
- $\sum U_i$: le nombre de clients livrés en retard.

REFERENCES

Bouly, H. , D-C. Dang et A. Moukrim, 2010. A memetic algorithm for the team orienteering problem. *4OR*, 8(1), p49-70.

Cattaruzza D. , N. Absi, D. Feillet, T. Vidal, 2013. A Hybrid Genetic Method for the Multi Trip Vehicle Routing Problem, *ROADEF 2013 (14ème congrès de la société française de Recherche Opérationnelle et d'Aide à la Décision).*

Chen H-K., C-F. Hsueh, et M-S. Chang, 2009. Production scheduling and vehicle routing with time windows for perishable food products. *Computers and Operations Research*, 36, p. 2311-2319.

Chen Z.L., et G.L. Vairaktarakis, 2005. Integrated scheduling of production and distribution operations. *Management Science*, 54, p. 614-628.

Kergosien, Y., J-F. Tournamille, B. Laurence et J.-C. Billaut, 2011. Planning and tracking chemotherapy production for cancer treatment: a performing and integrated solution. *International Journal of Medical Informatics*, 80, p. 655-662.

Lee J., B. Kim, A. Johnson et K. Lee, 2014. The nuclear medicine production and delivery problem. *European Journal of Operational Research*, 236, p. 461-472.

Li C-L., G. Vairaktarakis, et C-Y. Lee, 2005. Machine scheduling with deliveries to multiple customer locations. *European Journal of Operational Research*, 164, p. 39-51.

Olivera, A. et O. Viera, 2007. Adaptive memory programming for the vehicle routing problem with multiple trips.*Computers and Operations Research*, 34, P. 28-47.

Macedo, R., C. Alves, J.M.V. de Carvalho, F. Clautiaux et S. Hanafi, 2011. Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research*, 214, p. 536-545.

Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research*, 31, p. 1985-2002.

Toptal A., U. Koc, et I. Sabuncuoglu, 2014. A joint production and transportation planning problem with heterogeneous vehicles. *Journal of the Operational Research Society*, 65, p. 180-196.

Tsirimpas, P., A. Tatarakis, I. Minis et E.G. Kyriakidis, 2008. Single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research*, 187, p. 483-495.