

Université de Montréal

**Planification de tournées de véhicules pour le
problème de livraison à domicile**

11507740

par

Nabila Azi

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.) en informatique

Avril 2004

© Nabila Azi, 2004.



QA

76

U54

2004

V.026

Direction des bibliothèques

AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :
**Planification de tournées de véhicules pour le
problème de livraison à domicile**

présenté par :

Nabila Azi

a été évalué par le jury d'examen composé des personnes suivantes :

M. Michael Florian

Président-rapporteur

Jean-Yves Potvin

Directeur de recherche

Michel Gendreau

Co-directeur de recherche

Patrice Marcotte

Membre du jury

Mémoire accepté le :

21 mai 2004

SOMMAIRE

Nous considérons le problème de tournées de véhicules avec fenêtres de temps et utilisation multiple de véhicules. C'est une variante du problème classique de tournées de véhicules où un même véhicule peut être réutilisé plusieurs fois afin d'effectuer plusieurs tournées de livraison sur l'horizon de planification. De plus, chaque client dans une tournée doit être visité avant une limite de temps donnée. Un revenu est associé à chaque client et le revenu d'une tournée, débutant et se terminant au dépôt, est défini comme étant la somme des revenus des clients visités dans la tournée. Le problème consiste alors à déterminer un ensemble de tournées satisfaisant les contraintes et maximisant le revenu total moins les coûts de transport. Ce problème s'avère très intéressant en pratique pour la livraison à domicile de produits périssables, des denrées alimentaires par exemple, où il est important d'effectuer plusieurs tournées de courte durée et de les combiner afin de constituer une journée de travail.

Notre approche de résolution est une méthode exacte basée sur une modélisation du problème sous forme de graphe dans lequel on recherche des plus courts chemins sous des contraintes de ressources. Pour notre problème de livraison avec tournées multiples, où l'on maximise le revenu associé aux clients desservis, ce plus court chemin doit impérativement être élémentaire, c'est à dire ne passer qu'une fois au plus à chaque client.

Mots-clés : Problème de livraison à domicile, utilisation multiple de véhicules, plus courts chemins élémentaires, denrées périssables.

ABSTRACT

We consider a variant of the standard vehicle routing problem where the same vehicle may be assigned to several routes during a given planning period. Furthermore, a revenue is associated with each client. The revenue of a route is the sum of revenues of all clients in the route. The objective is to maximise the revenue minus the routing costs, subject to various constraints. This problem has received little attention in the literature in spite of its importance in practice. For example, the home delivery of perishable goods, like food, implies that routes are of short duration and must be combined to form a complete working day. We believe that this type of problem will become increasingly important in the future with the advent of electronic services, like e-groceries, where customers can order goods through the Internet and get them delivered at home. The problem is addressed via an exact approach that exploits an elementary shortest path algorithm with resource constraints, like capacity and time.

Keywords: home delivery problem, multiple use of vehicles, elementary shortest path, perishable goods.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	vii
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
INTRODUCTION	1
CHAPITRE 1 : REVUE DE LITTÉRATURE	4
1.1 : Problème de tournées de véhicules avec réutilisation de véhicules	6
1.2 : Problème de livraison à domicile	7
1.3 : Services d'épicerie en ligne	11
1.3.1 : Comparaison des services offerts	11
CHAPITRE 2 : MODÉLISATION	16

2.1 : Problème avec un seul véhicule	16
2.1.1 : Notations	16
2.1.2 : Variables	17
2.1.3 : Modèle	19
2.2 : Problème avec plusieurs véhicules	21
2.2.1 : Modèle	22
CHAPITRE 3 : PROBLÈME DU PLUS COURT CHEMIN ÉLÉMENTAIRE AVEC CONTRAINTES	24
3.1 : Introduction	24
3.2 : Complexité du problème	25
3.3 : Algorithmes de Plus Court Chemin	25
3.4 : Plus Courts Chemins avec Contraintes de Ressources	28
3.4.1 : Approches par étiquetage	28
3.4.2 : Approche de Feillet et al.	30
CHAPITRE 4 : MÉTHODE DE RÉOLUTION	35
4.1 : Phase 1 - Génération des tournées avec contraintes de ressources	36
4.2 : Phase 2 - Construction du graphe des tournées	38
4.2.1 : Temps de début et de fin au plus tard d'une tournée	39
4.2.2 : Temps de début et de fin au plus tôt d'une tournée	40
CHAPITRE 5 : EXPÉRIMENTATION	45

5.1 : Effet du critère de dominance en phase 1	46
5.2 : Résultats sur les instances de Solomon de type 2	47
5.3 : Résultats sur les instances générées à partir de tournées individuelles . . .	56
CONCLUSION	60
BIBLIOGRAPHIE	62

LISTE DES TABLEAUX

Tableau 1.1 : Comparaison de services de détaillants en ligne	12
Tableau 4.1 : Fenêtres de temps associées aux tournées.	43
Tableau 5.1 : Moyenne du nombre de tournées générées en phase 1.	47
Tableau 5.2 : Solutions pour les instances R avec $t_{max} = 75$	49
Tableau 5.3 : Solutions pour les instances R avec $t_{max} = 90$	50
Tableau 5.4 : Solutions pour les instances RC avec $t_{max} = 75$	51
Tableau 5.5 : Solutions pour les instances RC avec $t_{max} = 90$	52
Tableau 5.6 : Solutions du problème pour les instances C avec $t_{max} = 220$. . .	53
Tableau 5.7 : Solutions du problème pour les instances C avec $t_{max} = 240$. . .	54
Tableau 5.8 : Solutions pour les instances RC	58
Tableau 5.9 : Solutions pour les instances R	59

LISTE DES FIGURES

Figure 4.1 : Une tournée partielle r	37
Figure 4.2 : Une tournée partielle r'	38
Figure 4.3 : Graphe de tournées	44

À Mounir...

REMERCIEMENTS

Je voudrais remercier mes directeurs, les professeurs Michel Gendreau et Jean-Yves Potvin, de m'avoir proposé ce sujet et pour le soutien financier qu'ils m'ont accordé. Les congrès auxquels ils m'ont permis d'assister m'ont été très bénéfiques et m'ont permis de rencontrer beaucoup de personnes. Je tiens à remercier le professeur Jean-Yves Potvin pour sa disponibilité et pour m'avoir guidée tout au long de mon travail par ses remarques éclairées. Je tiens également à le remercier pour toute la confiance qu'il m'a témoignée. Je voudrais exprimer mes plus sincères remerciements au professeur Michel Gendreau, pour son support, ses encouragements, et pour m'avoir fait bénéficier de son expérience dans ce domaine de recherche.

Mes remerciements vont aussi aux membres de jury.

Je tiens également à remercier les personnes qui travaillent au Centre de Recherche sur les Transports, pour leur accueil et les moyens mis à ma disposition.

INTRODUCTION

Dans un contexte économique toujours plus concurrentiel, l'optimisation des ressources prend une place très importante dans les prises de décision des entreprises, y compris celles évoluant dans le domaine des transports. Des efforts considérables sont donc entrepris pour réduire les coûts et la consommation des différentes ressources, qu'elles soient matérielles ou humaines.

Le problème de tournées de véhicules à moindre coût est justement un problème logistique important pour la distribution de biens et le transport de personnes. En effet, plusieurs applications réelles, telles le transport scolaire, le transport adapté et les services de courrier entraînent la confection de tournées de véhicules. Nous nous intéressons plus particulièrement dans ce mémoire au problème de tournées de véhicules avec fenêtres de temps et utilisation multiple de véhicules. C'est une extension du problème classique de tournées de véhicules avec fenêtres de temps où une fenêtre est associée à chaque client pour contraindre le temps de début de service. Ici, toutefois un même véhicule peut effectuer plusieurs tournées différentes sur l'horizon de planification.

Le problème de tournées avec réutilisation de véhicules a été très rarement étudié dans la littérature bien que l'on retrouve maintenant de nombreuses applications dans le monde réel. Ainsi, avec l'avènement du commerce électronique, on constate aujourd'hui une prolifération des magasins en ligne qui offrent des services de livraison à domicile, soit le dernier kilomètre (*last mile*) à parcourir pour atteindre le client. Dépendamment du contexte, ce dernier peut être présent au moment de la livraison (service attendu) ou non (service non

attendu). Le commerce électronique est devenu incontournable dans des secteurs comme les voyages (transport, hôtels), les biens informatiques (logiciels, ordinateurs), les biens culturels (livres, musique), ainsi que dans le secteur des épiceries électroniques qui se caractérisent par la distribution de produits alimentaires périssables. Dans ce dernier cas, il est important d'effectuer plusieurs tournées de courte durée afin de préserver la qualité des produits livrés. Le problème de tournées de véhicules avec utilisation multiple de véhicules s'adapte donc très bien à ce type d'applications.

Notre étude se concentre principalement sur le développement d'une méthode exacte basée sur la modélisation du problème sous forme de graphe dans lequel on recherche des plus courts chemins avec contraintes de capacité et de temps. Pour notre problème de livraison avec tournées multiples, où l'on doit choisir un sous-ensemble de clients à desservir maximisant le revenu total, ce plus court chemin doit impérativement être élémentaire, c'est à dire ne passer qu'une fois au plus à chaque client.

Le mémoire est organisé comme suit. Nous présenterons au premier chapitre une revue de la littérature consacrée aux problèmes de livraison à domicile, incluant une étude comparative de certains détaillants en ligne, comme les épiceries électroniques. Dans le deuxième chapitre, nous présenterons une formulation mathématique du problème de tournées de véhicules avec fenêtres de temps et utilisation multiple de véhicules, d'abord dans le cas d'un véhicule, et ensuite dans le cas de plusieurs véhicules. Etant donné que le problème de la recherche d'un plus court chemin est à la base de notre approche de résolution, le chapitre 3 sera consacré à la description d'un algorithme de plus courts chemins avec contraintes de ressources. Le chapitre 4 décrira ensuite l'algorithme que nous avons mis en oeuvre pour la résolution de notre problème, dans le cas d'un seul véhicule. Des résultats expérimentaux

seront présentés au chapitre 5. Pour finir, la conclusion présentera quelques perspectives pour des recherches futures.

Chapitre 1

Revue de littérature

Le problème de tournées de véhicules est un problème central dans la gestion de la distribution des biens et dans le transport des personnes. Le problème de tournées de véhicules avec fenêtres de temps et utilisation multiple de véhicules est une extension du problème classique de tournées de véhicules où une fenêtre de temps est associée à chaque client pour contraindre le temps de début de service (Bräysy et Gendreau, 2001). Ici, toutefois, un même véhicule peut réaliser plusieurs tournées sur l'horizon de planification. Ce problème a été très rarement étudié en littérature sauf dans (Fleischmann, 1990 ; Taillard et al., 1995).

Avec le développement du commerce électronique, on assiste aujourd'hui à une prolifération des magasins en ligne qui offrent un service de livraison à domicile pouvant être modélisé comme un problème de tournées avec réutilisation de véhicules. Par exemple, grâce à l'Internet, il est maintenant possible de parcourir, depuis chez soi, les rayonnages d'un supermarché virtuel, et de cliquer sur les fruits et légumes et autres produits alimentaires que l'on souhaite recevoir.

Le service de livraison à domicile n'est cependant pas un nouveau concept. Ce service était déjà offert auparavant par les épicerie et les pharmacies (entre autres). Mais, avec

l'introduction de l'Internet dans les foyers, ce service est devenu de plus en plus répandu, et nécessaire pour les personnes à mobilité réduite et les personnes qui ne veulent pas passer beaucoup de temps dans les magasins.

Les entreprises fournissant des services d'épicerie électronique ont des marges bénéficiaires très étroites. C'est ainsi que seules quelques-unes ont survécu à la faillite et sont encore profitables. Une question critique ici est comment charger un prix adéquat pour la livraison compte tenu des coûts d'opérations et du désir de servir les clients dans l'intervalle de temps le plus convenable pour eux.

Bien que les sites Webvan.com, HomeRuns.com et Streamline.com ont fait faillite, on observe un retour des magasins en ligne dans le marché alimentaire. Les analystes sont optimistes quand à l'avenir de ce secteur. Ainsi, le distributeur britannique Tesco est aujourd'hui considéré comme le leader mondial de l'épicerie sur Internet et constitue un modèle du genre pour ses concurrents. Il couvre actuellement 95 % du territoire britannique et 75 % de la population irlandaise. Aux Etats-Unis, il s'est appuyé sur le groupe de distribution Safeway pour développer ses activités. Son arrivée en Corée constitue aussi une première sur le marché asiatique.

Nous allons maintenant présenter le problème de tournées de véhicules avec réutilisation des véhicules tel qu' étudié par Taillard et al. (1996). Par la suite, nous présenterons deux travaux qui ont été effectués dans le contexte de la livraison à domicile.

1.1 Problème de tournées de véhicules avec réutilisation de véhicules

Taillard et al. (1996) proposent une heuristique pour la résolution du problème de tournées de véhicules avec utilisation multiple de véhicules, mais sans fenêtres de temps. Cette heuristique se divise en trois phases. Dans la phase 1, un ensemble de routes pour le problème classique (sans réutilisation de véhicules) est généré en utilisant un algorithme de recherche tabou (Rochat et Taillard 1995).

Dans la phase 2, on sélectionne un sous-ensemble de routes générées par la phase 1, en favorisant les routes appartenant à de bonnes solutions. A partir de ce sous-ensemble, on génère l'ensemble K de toutes les solutions possibles au problème classique de tournées de véhicules par une approche énumérative.

Dans la phase 3, on tente d'obtenir une solution réalisable au problème avec réutilisation de véhicules, en résolvant un problème de mise en boîte (*bin-packing*) pour chaque solution k de l'ensemble K , où $f_{k,l}$ est la durée de la l^{me} route pour $l = 1, \dots, m_k$, et m_k est le nombre de routes dans la solution k . Une solution consiste à placer les m_k articles (routes) de poids $f_{k,1}, \dots, f_{k,m_k}$ dans m boîtes de taille M , où m est le nombre de véhicules disponibles ($m \ll m_k$) et M est la durée maximale d'une journée de travail pour un véhicule. Pour identifier de telles solutions, les articles sont considérés en ordre croissant de leur poids et sont affectés à tour de rôle à la boîte dont le poids accumulé est minimal. Si aucune des m boîtes n'a un poids accumulé qui dépasse M , une fois que tous les articles sont placés, alors on a une solution. Sinon, on échange des articles appartenant à différentes boîtes dans

l'espoir de trouver une solution. Cependant, il n'est pas garanti qu'une solution réalisable soit trouvée de cette façon. On peut donc étendre les solutions admissibles au domaine non réalisable, en permettant d'excéder la durée maximale d'une journée de travail, moyennant une pénalité dans l'objectif.

1.2 Problème de livraison à domicile

Campbell et Savelsbergh (2002) s'intéressent au problème de livraison à domicile dans un contexte dynamique où les requêtes arrivent en temps réel et une décision d'accepter ou non la requête doit être prise sur le champ. Les routes ne sont toutefois exécutées que bien après que les décisions d'accepter ou de rejeter les requêtes aient été prises.

Dans ce travail, on suppose que les clients potentiels sont connus à l'avance et que chaque client a une probabilité p_i d'appeler. Cette probabilité décroît linéairement avec le temps jusqu'à la fin de l'horizon de planification. A chaque commande d'un client est associée une demande d_i et un revenu r_i . L'objectif est de maximiser le profit, i.e. le revenu total moins les coûts de livraison. Les auteurs résolvent un problème de tournées de véhicules à chaque fois qu'une nouvelle requête se matérialise, en incluant les requêtes déjà acceptées, la nouvelle requête et les requêtes anticipées. Si le profit espéré sans cette requête est plus grand que le profit avec la requête, alors cette dernière est rejetée, sinon elle est acceptée dans l'intervalle de temps associé au plus grand profit espéré. Pour les requêtes futures (i.e., non encore matérialisées), les demandes et les revenus sont ajustés selon leurs probabilités.

Les auteurs proposent différents critères d'insertion pour estimer le coût espéré d'une route sans la requête et avec la requête. L'estimation des coûts d'insertion d'une requête future

j entre les requêtes $i - 1$ et i se fait selon les critères suivants (où r_j est le profit associé à la requête j et c_{ij} est le coût de l'arc (i, j)) :

- a. profit simple : $r_j - (c_{i-1,j} + c_{ji} - c_{i-1,i})$,
- b. profit actualisé : $p_j r_j - (c_{i-1,j} + c_{ji} - c_{i-1,i})$

Dans les deux critères précédents, les requêtes $i - 1$ et i peuvent être des requêtes déjà acceptées ou non. Pour tenir compte du fait que les requêtes futures peuvent ne jamais se réaliser, d'autres critères ont été proposés :

- c. Soient u et v deux requêtes déjà acceptées. La contribution d'une arête (j, k) au coût espéré du segment de route compris entre u et v étant :

$$c_{j,k} p_j p_k \prod_{l=j+1, \dots, k-1} (1 - p_l)$$

Nous obtenons le critère suivant pour l'insertion de j entre u et v

$$\sum_{j=u, \dots, v-1} \sum_{k=u+1, \dots, v} c_{j,k} p_j p_k \prod_{l=j+1, \dots, k-1} (1 - p_l)$$

où c_{jk} est le coût de l'arête (j, k) , p_j et p_k sont les probabilités de réalisation des requêtes j et k , et $(1 - p_l)$ est la probabilité de non occurrence de la requête l .

- d. Ici, on estime d'abord le coût d'insertion v_1 d'une requête j entre deux sommets voisins quelconques $i - 1$ et i , puis le coût v_2 entre deux sommets voisins déjà acceptés $k - 1$ et k :

$$v_1 = c_{i-1,j} + c_{j,i} - c_{i-1,i}$$

$$v_2 = c_{k-1,j} + c_{j,k} - c_{k-1,k}$$

Etant donné que le coût de base est v_2 , et que la différence entre v_2 et v_1 , pondérée par les probabilités de présence des requêtes $i - 1$ et i , est $p_{i-1}p_i(v_2 - v_1)$, le profit réalisé en insérant j est estimé par :

$$p_j r_j - (v_2 - p_{i-1}p_i(v_2 - v_1)).$$

Dans Lin et Mahmassani (2002), les auteurs examinent les compromis qui existent entre le modèle de distribution centralisé (*warehouse*) et le modèle décentralisé (*brick-and-click*). Dans le modèle centralisé, le commerçant opère avec un ou plusieurs entrepôts pour faire les livraisons. Souvent, ce sont des sites dédiés à ces activités, mais le nombre de demandes doit être suffisamment grand pour justifier un tel investissement. Lorsque le client choisit un ou plusieurs articles, sa commande est transmise par le supermarché en ligne à l'entrepôt le plus proche du domicile du consommateur. Dans le modèle décentralisé, le distributeur peut posséder un ou plusieurs entrepôts mais opère aussi en partenariat avec des centres d'achats déjà existants pour livrer ses commandes. Une fois que le consommateur en ligne aura soumis sa commande, celle-ci sera éclatée en sous-commandes selon les produits.

Selon Lin et Mahmassani (2002), les facteurs logistiques importants dans la viabilité du commerçant en ligne et dans le niveau de service offert au client sont :

- Le nombre de dépôts
- La topologie du réseau de distribution
- Le nombre de véhicules qui opèrent à chaque dépôt.
- La variabilité dans les demandes et dans les fenêtres de temps

Les auteurs étudient aussi différents scénarios de livraison à domicile, soit :

Scénario 1 - Une fenêtre de temps identique pour tous les clients (service non attendu).

Scénario 2 - Différentes fenêtres de temps, mais de même durée.

Scénario 3 - Différentes fenêtres de temps de durées différentes.

Dans le scénario 1, il y a une seule fenêtre de temps qui correspond à une journée entière. Des compagnies telles que ShopLink.com et Streamline.com adoptent cette politique où les clients n'ont pas à choisir une fenêtre de temps. En effet, les compagnies installent des réfrigérateurs dans les immeubles de telle sorte que le client n'a pas besoin d'être à la maison pour recevoir sa livraison. Il n'y a pas de frais de livraison mais un montant à payer à chaque mois.

Le scénario 2 est similaire à celui de Peapod.com et Albertsons.com où les fenêtres de temps durant la journée ont la même durée. Par exemple, la journée est divisée en intervalles de 90 minutes qui ne se chevauchent pas et le client doit choisir l'intervalle qui lui convient. Le nombre maximum de clients que l'on peut servir par intervalle est fixé en fonction de la longueur de l'intervalle.

Dans le scénario 3, différentes durées pour les fenêtres de temps sont proposées aux clients. Des simulations avec des durées de 2 heures, 4 heures et 6 heures, ont montré que les coûts de livraison sont les moins élevés avec ce scénario (qui est en fait une combinaison des deux scénarios précédents).

1.3 Services d'épicerie en ligne

L'essor du commerce électronique s'est accompagné d'une intense activité de recherches, se traduisant par de nombreuses études à la fois théoriques et empiriques. ECOMLOG (2001) est un groupe de chercheurs de l'université de Helsinki qui ont fait une étude détaillée, basée sur des simulations, de différents concepts associés aux services de détaillants en ligne. Ils ont produit, entre autres, plusieurs articles relatifs aux épicerie en ligne. Ainsi, Punakivi et Saranen (2001), Punakivi (2001), Kämäräinen, Saranen et Holmström (2001) et Yrjola (2001) ont étudié différentes façons de recevoir, traiter et livrer les commandes, et leurs effets sur les coûts.

1.3.1 Comparaison des services offerts

Le tableau 1.1 résume les caractéristiques de quelques épicerie en ligne comme Webvan.com, Peapod.com, HomeRuns.com, Streamline.com, Shoplink.com, Netgrocer.com, Alberstons.com, et Tesco.com. Le service offert par ces détaillants se caractérise par :

- 1) les coûts de livraison,
- 2) les heures et les délais de livraison,
- 3) les durées des fenêtres de temps
- 4) le modèle de distribution.

Chaque détaillant en ligne a sa propre politique de gestion, que ce soit pour les délais ou les coûts de livraison. Dans la plupart des cas, le client doit finaliser sa commande au moins un jour avant la livraison. Par exemple, Peapod.com demande à ses clients de finaliser leurs

Compagnies	Produits	Tarifs de livraison	Horaires Délais de Livraison	Année Ouverture	Année Fermeture
Peapod.com	Périssables	Gratuit (>75\$) 9.95\$ (<75\$) minimum(50\$)	07 :00-13 :00 16 :00-21 :00 2 Heures (Jour suivant)	1989	
Netgrocer.com	Non-Périssables	4.99\$ à 5.99\$ (Dépend de la zone et taille de la commande)	FedEx (2-3 jours)	1995	
Alberstons.com	Périssables	9.95\$ fixe	10 :00-21 :00 90 min. Même jour, après 17 :00, pour les commandes placées avant 10 :00 et jour suivant pour celles entre 10 :00et minuit.	1999	
Webvan.com	Périssables	9.95\$(<75\$) 4.95\$ (75-100\$) Gratuit(>100\$)	06 :00-22 :00 30 min. (même jour)	1999	Juillet 2001
HomeRuns.com	Périssables	2.50\$	07 :00-23 :00 Différentes durées		Juillet 2001
Streamline.com	Périssables	30\$ 4 fois/Semaine	09 :00-18 :00 Jour fixe (non attendu)	1993	Novembre 2000
Shoplink.com	Périssables	35\$ 4 fois/Semaine	09 :00-18 :00 Jour fixe (non attendu)	1996	Novembre 2000
Tesco.com	Périssables	9.95\$ fixe	10 :00-21 :00 2 heures (Jour Suiv.)		

Tableau 1.1 – Comparaison de services de détaillants en ligne

commandes avant minuit afin d'être livrés le lendemain. La plupart des détaillants ont leur propre flotte de véhicules, à part NetGrocer.com, qui utilise le service de FedEx pour livrer ses commandes.

Tel que mentionné plus tôt, les commandes peuvent être livrés à partir d'un centre de distribution spécialement créé pour le service ou à partir d'un centre d'achats déjà existant. Dans ce dernier cas, la livraison à domicile est moins coûteuse et requiert un capital initial moins important (Yrjola 2001). Yrjola, Tanskanen, et Holmström (2002) prétendent toutefois que le modèle du centre d'achat n'est pas efficace opérationnellement. Ils suggèrent de commencer par ce modèle pour éviter un grand investissement au départ et d'adopter par la suite le modèle d'un centre de distribution. Webvan.com est un exemple du modèle de centre de distribution dédié à la livraison. Par contre, Tesco.com et Peapod.com suivent le modèle d'un centre d'achats existant.

Un autre aspect important est la durée des fenêtres de temps. Par exemple, dans Webvan, les clients choisissent une fenêtre de temps de 30 minutes et il y a une pénalité de retard pour le livreur et une autre pour le client s'il n'est pas présent au moment de la livraison. Webvan était le seul à offrir le service de livraison le même jour, et il a fermé en juillet 2001. Evidemment, plus la fenêtre de temps est serrée, plus les coûts de livraison sont élevés. Saranen et Småros (2001) ont fait des simulations avec deux modèles spécifiques, Streamline.com qui adopte le service non attendu et Webvan.com qui adopte le service attendu avec un délai de livraison d'une demi-heure. Ils montrent que les coûts de livraison de Webvan.com sont cinq fois plus élevés. Le service non attendu diminue considérablement les coûts en permettant de choisir la fenêtre de temps la plus profitable, ce qui mène à des tournées de véhicules plus efficaces. Pour le service non attendu, où

l'aspect des fenêtres de temps n'est pas important, Punakivi (2001) étudie l'utilisation de routes fixes versus l'utilisation de routes optimisées. La simulation a révélé que les diminutions de coûts avec l'utilisation de routes optimisées sont de l'ordre de 18% pour les régions denses et vont jusqu'à 54% pour les régions non denses. Punakivi (2001) propose aussi d'effectuer des tournées de durée limitée et d'utiliser au maximum la capacité des véhicules. Il montre que plus le client a le contrôle sur la sélection de sa fenêtre de temps pour la livraison, plus les coûts des opérations sont élevés.

Un autre aspect à prendre en considération, est la tarification des livraisons. Initialement, les livraisons étaient gratuites peu importe la quantité commandée. Par la suite, on chargeait seulement les livraisons dont la valeur ne dépassait pas un certain montant. Maintenant, les frais de livraison varient souvent selon la plage à laquelle appartient la valeur de la marchandise commandée. Par exemple, le coût de livraison chez Webvan.com est de 9.95 \$ pour les commandes de moins de 75 \$ et il n'y a pas de frais pour les commandes de plus de 100 \$. Un autre exemple est Peapod.com. Sa structure de tarification est similaire à celle de Webvan, mais le montant minimum requis pour être livré est de 50\$.

Afin de survivre aux fluctuations du marché et à la concurrence, les commerces d'épicerie en ligne se voient donc confrontés à plusieurs problèmes logistiques dans la distribution des marchandises. En dépit des défis qui se posent aux services en ligne, les commerçants peuvent en tirer plusieurs avantages. Entre autres, par l'utilisation du Web, ils peuvent avoir un historique de leurs clients et de leurs achats, proposer des promotions à leurs clients et leur offrir des rabais afin de les encourager à accepter la fenêtre de temps qu'on leur propose (ce qui est plus profitable pour le commerçant).

Cette étude bibliographique a fait un bilan de la situation qui existe dans le domaine de l'épicerie électronique et, compte tenu des marges bénéficiaires étroites de ce secteur, à

quel point il est important pour le commerçant de bien sélectionner les clients à desservir (en fonction des revenus et coûts associés) afin d'augmenter ses profits.

Puisqu'un revenu est associé à chaque client, le problème de livraison à domicile appartient aussi à la classe des problèmes de tournées de véhicules avec gains, où l'objectif est d'optimiser une fonction tenant compte du revenu total accumulé et des coûts d'opérations (Feillet, et al. 2001). Dans le chapitre suivant, nous présenterons une formulation générale du problème de livraison à domicile avec réutilisation des véhicules, pour le cas où un seul véhicule est disponible et pour le cas où plusieurs véhicules sont disponibles.

Chapitre 2

Modélisation

Dans ce chapitre, nous présentons une formulation mathématique pour le problème de tournées de véhicules avec fenêtres de temps et utilisation multiple de véhicules. Nous supposons qu'un revenu est associé à chaque client. Nous nous intéressons d'abord au cas où un seul véhicule est disponible. Par la suite, nous étudions le cas où plusieurs véhicules sont disponibles.

2.1 Problème avec un seul véhicule

2.1.1 Notations

Soit un véhicule devant desservir un ensemble de clients à partir d'un dépôt central dans un graphe orienté. Le graphe contient $|N| + 2$ sommets, où les clients sont dénotés par $1, 2, \dots, n$ et le dépôt est représenté par les sommets 0 et $n + 1$ selon qu'il s'agisse du noeud initial ou terminal d'un arc. L'ensemble des sommets est dénoté par $N^+ = N \cup \{0, n + 1\}$

et l'ensemble des arcs $A^+ = A \cup \{(0, n+1)\}$, où A est l'ensemble des arcs dans le graphe original et $(0, n+1)$ est un arc fictif qui est utilisé pour représenter une tournée vide. A chaque arc $(i, j) \in A^+$ est associé une distance d_{ij} et un temps de parcours t_{ij} , avec $d_{0(n+1)} = 0$ et $t_{0(n+1)} = 0$. Le véhicule a une capacité Q et chaque client i a une demande q_i , un revenu p_i , un temps de service s_i et une fenêtre de temps $[a_i, b_i]$ (avec $s_0 = s_{n+1} = 0$, $q_0 = q_{n+1} = 0$, $a_0 = a_{n+1} = 0$ et $b_0 = b_{n+1} = \infty$). Le véhicule doit toujours arriver au client i avant b_i . Il peut arriver avant a_i , mais dans ce cas, il doit attendre.

Une journée de travail est formée par l'ensemble des tournées $K = \{1, 2, \dots, k\}$ où chaque tournée commence et se termine au dépôt (certaines de ces tournées peuvent être vides). De plus, nous supposons, sans perte de généralité, que les tournées sont numérotées dans l'ordre $1, 2, \dots, k$.

A cause de la présence de denrées périssables, chaque client doit être servi avant un certain délai, défini par une constante t_{max} qui est additionné au temps de début de la tournée. Aussi, on associe à chaque tournée r , un temps de chargement σ_r .

2.1.2 Variables

Trois types de variables de décision sont introduites dans le modèle mathématique :

- Pour tout $r \in K$, et pour tout arc $(i, j) \in A^+$, la variable binaire x_{ij}^r est définie comme suit :

$$x_{ij}^r = \begin{cases} 1 & \text{si l'arc } (i, j) \text{ appartient à la tournée } r \\ 0 & \text{sinon} \end{cases}$$

avec $x_{0(n+1)}^r = 1$ si la tournée r est vide.

b. Pour tout $r \in K$ la variable y_i^r est définie par :

$$y_i^r = \begin{cases} 1 & \text{si le client } i \text{ est dans la tournée } r \\ 0 & \text{sinon} \end{cases}$$

$$y_i^r = \sum_{j \in N^+} x_{ij}^r \quad \forall i \in N$$

c. Pour chaque client $i \in N$, la variable $t_i^r \in \mathbb{R}^+$ indique le moment où le service au noeud i commence dans la tournée r . Pour chaque tournée $r \in K$, t_0^r est l'instant de début de la tournée r et t_{n+1}^r l'heure de fin de la tournée r .

2.1.3 Modèle

Le modèle se formule comme suit (où M est une constante très grande).

$$\text{Min} \sum_{(i,j) \in A} d_{ij} x_{ij}^r - \alpha' \sum_{i \in N} p_i y_i^r \quad (2.1)$$

$$\text{sujet à :} \quad \sum_{j \in N^+} x_{ij}^r = y_i^r, \quad i \in N, r \in K, \quad (2.2)$$

$$\sum_{r \in K} y_i^r \leq 1, \quad i \in N, \quad (2.3)$$

$$\sum_{i \in N^+} x_{ih}^r - \sum_{j \in N^+} x_{hj}^r = 0, \quad h \in N, r \in K, \quad (2.4)$$

$$\sum_{i \in N^+} x_{0i}^r = 1, \quad r \in K, \quad (2.5)$$

$$\sum_{i \in N^+} x_{i(n+1)}^r = 1, \quad r \in K, \quad (2.6)$$

$$\sum_{i \in N} q_i y_i^r \leq Q, \quad r \in K, \quad (2.7)$$

$$t_i^r + s_i + t_{ij} - M(1 - x_{ij}^r) \leq t_j^r, \quad (i, j) \in A^+, r \in K, \quad (2.8)$$

$$a_i y_i^r \leq t_i^r \leq b_i y_i^r, \quad i \in N, r \in K \quad (2.9)$$

$$t_0^1 \geq \sigma^1, \quad (2.10)$$

$$t_{n+1}^r + \sigma^{r+1} \leq t_0^{r+1}, \quad r = 1, \dots, k-1, \quad (2.11)$$

$$t_i^r \leq t_0^r + t_{max}, \quad i \in N, r \in K, \quad (2.12)$$

$$\sigma^r = \beta \sum_{i \in N} s_i y_i^r, \quad r \in K, \quad (2.13)$$

$$x_{ij}^r \text{ binaire}, \quad (i, j) \in A^+, r \in K, \quad (2.14)$$

$$y_i^r \text{ binaire}, \quad i \in N, r \in K \quad (2.15)$$

L'objectif est de trouver un ensemble de tournées réalisables minimisant la distance totale parcourue moins la somme des revenus associés aux clients desservis. La composante revenu de la fonction objectif est pondérée par un paramètre α' . Avec α' suffisamment grand et un revenu unitaire associé à chaque client, l'objectif correspond alors à maximiser le nombre des clients à desservir, et pour un même nombre de clients, à minimiser la distance totale parcourue.

Contraintes

- La contrainte (2.2) fait le lien entre les variables de flots x_{ij}^r et les variables t_i^r .
- La contrainte (2.3) assure que chaque client est visité au plus une fois.
- La contrainte (2.4) garantit la propriété de conservation de flot.
- Les contraintes (2.5) et (2.6) assurent que chaque tournée commence et se termine au dépôt.
- La contrainte (2.7) assure que la capacité du véhicule n'est pas dépassée dans chaque tournée.
- Avec la contrainte (2.8), un véhicule ne peut arriver à j avant $s_i + t_{ij}$ s'il se déplace de i à j .
- La contrainte (2.9) assure que les fenêtres de temps sont respectées.
- Les contraintes (2.10) et (2.11) stipulent que la première tournée ne peut débiter avant son chargement et assurent aussi la continuité des tournées.
- La contrainte (2.12) stipule que chaque client soit servi à l'intérieur du délai maximal.
- Enfin, la contrainte (2.13) indique que le temps de chargement d'un véhicule est égal à la somme des temps de service des clients à desservir pondérée par un paramètre β .

Notons que la contrainte (2.9) force les variables t_i^r à être nulles si le client i n'est pas dans la tournée r . Par conséquent, les contraintes (2.12) sont automatiquement satisfaites dans ce cas.

2.2 Problème avec plusieurs véhicules

Nous reprenons ci-dessous la modélisation du problème, mais cette fois avec plusieurs véhicules. On suppose que l'on a V véhicules. Afin de ne pas alourdir la formulation suite à l'ajout d'un nouvel indice, nous introduisons un autre type de variable de décision z_{rl} . Soit K l'ensemble des tournées de tous les véhicules. Pour tout $r \in K$ et $l \in K$, $r < l$, la variable binaire z_{rl} est définie comme suit :

$$z_{rl} = \begin{cases} 1 & \text{si la tournée } l \text{ suit immédiatement la tournée } r \text{ dans} \\ & \text{la journée de travail d'un même véhicule} \\ 0 & \text{sinon.} \end{cases}$$

Sans perte de généralité, on suppose que les tournées apparaissent dans l'ordre croissant de leurs indices dans toute journée de travail d'un véhicule. Donc, les variables z_{rl} n'existent que pour $r < l$.

2.2.1 Modèle

$$\text{Min} \sum_{r \in K} \sum_{(i,j) \in A} d_{ij} x_{ij}^r - \alpha' \sum_{i \in N} p_i y_i^r \quad (2.16)$$

$$\text{sujet à :} \quad \sum_{j \in N^+} x_{ij}^r = y_i^r, \quad i \in N, r \in K, \quad (2.17)$$

$$\sum_{r \in K} y_i^r = 1, \quad i \in N, \quad (2.18)$$

$$\sum_{i \in N^+} x_{ih}^r - \sum_{j \in N^+} x_{hj}^r = 0, \quad h \in N, r \in K, \quad (2.19)$$

$$\sum_{i \in N^+} x_{0i}^r = 1, \quad r \in K, \quad (2.20)$$

$$\sum_{i \in N^+} x_{i(n+1)}^r = 1, \quad r \in K, \quad (2.21)$$

$$\sum_{i \in N} q_i y_i^r \leq Q, \quad r \in K, \quad (2.22)$$

$$t_i^r + s_i + t_{ij} - M(1 - x_{ij}^r) \leq t_j^r, \quad (i, j) \in A^+, r \in K, \quad (2.23)$$

$$a_i y_i^r \leq t_i^r \leq b_i y_i^r, \quad i \in N, r \in K, \quad (2.24)$$

$$t_0^r \geq \sigma^r, \quad r \in K, \quad (2.25)$$

$$t_i^r \leq t_0^r + t_{max}, \quad i \in N, r \in K, \quad (2.26)$$

$$\sigma^r = \beta \sum_{i \in N} s_i y_i^r, \quad r \in K, \quad (2.27)$$

$$t_0^l + M(1 - z_{rl}) \geq t_{n+1}^r + \sigma_l \quad r, l \in K, r < l, \quad (2.28)$$

$$\sum_{r \in K} \sum_{l \in K, l > r} z_{rl} \geq k - V \quad r, l \in K, \quad (2.29)$$

$$x_{ij}^r \in \{0, 1\} \quad \forall i, j \in A, r \in K, \quad (2.30)$$

$$y_i^r \in \{0, 1\} \quad \forall i \in C, r \in K, \quad (2.31)$$

$$z_{rl} \in \{0, 1\} \quad r, l \in K, r < l \quad (2.32)$$

Le modèle est identique au précédent, hormis l'ajout des deux contraintes suivantes :

- La contrainte (2.28) assure le séquençage des tournées d'un même véhicule. En effet, si la tournée l suit immédiatement la tournée r , alors $z_{rl} = 1$ et l'on a :

$$t_0^l \geq t_{n+1}^r + \sigma_l$$

Ainsi, la tournée l ne peut débuter avant la fin de la tournée r à laquelle on ajoute le temps de chargement de l . Autrement, la contrainte est automatiquement satisfaite à cause de la constante M qui est très grande.

- La contrainte (2.29) s'assure que si l'on a V véhicules, alors :

$$\sum_{r \in K} \sum_{l \in K, r < l} z_{rl} \geq k - V$$

où $\sum_{r \in K} \sum_{l \in K, r < l} z_{rl}$ est le nombre de transitions entre deux tournées consécutives dans la journée de travail des véhicules. Si l'on a un seul véhicule, on a $k - 1$ transitions. Si on a V véhicules, on a $k - V$ transitions, car il n'y a pas de transition entre la dernière tournée d'un véhicule et la première tournée du véhicule suivant.

Dans le chapitre suivant nous allons présenter un algorithme de recherche de plus court chemin élémentaire avec contraintes de ressources qui est à la base de notre approche de résolution dans le cas du problème avec un seul véhicule.

Chapitre 3

Problème du plus Court Chemin

Elémentaire avec Contraintes

3.1 Introduction

Le problème de recherche de plus court chemin élémentaire avec contraintes de ressources (dénomé par ESPRCP pour Elementary Shortest Path Resource Constrained Problem) apparaît fréquemment comme sous-problème dans la résolution de problèmes linéaires de grande taille lorsqu'on utilise la technique de génération de colonnes (e.g., les problèmes de confection de tournées de véhicules, de construction d'horaires d'équipages et autres problèmes similaires). Ce problème *NP-difficile* a donc déjà été étudié dans la littérature scientifique et nous présentons dans la suite un algorithme proposé dans Feillet et al. (2000) pour sa résolution.

3.2 Complexité du problème

Le problème du plus court chemin élémentaire avec contraintes de ressources est un problème *NP-difficile*. Pour le prouver, Larsen (1999) considère le cas particulier du chemin plus court élémentaire sans contraintes. La preuve stipule que s'il existait un algorithme polynomial pour le résoudre, alors il en existerait un pour résoudre le problème du circuit hamiltonien. Ce problème étant un problème *NP-Complet*, on aurait $P=NP$. Le problème du circuit hamiltonien est un problème de décision qui est défini sur un graphe orienté $G = (N, A)$. On doit alors déterminer si le graphe G contient un circuit hamiltonien.

Considérons une instance du problème de chemin plus court élémentaire donné par le graphe fortement connexe $G' = (N, A')$ obtenu à partir de $G = (N, A)$. Soit s un sommet quelconque défini comme étant le point de départ et d'arrivée du chemin. Associons le coût c_e suivant à chaque arc $e \in A'$

$$c_e = \begin{cases} -1 & \text{si l'arc } e \in A \\ 0 & \text{sinon} \end{cases}$$

Si la longueur du chemin est égale à $-|N|$, alors G contient un circuit hamiltonien.

3.3 Algorithmes de Plus Court Chemin

Une classe importante de méthodes pour résoudre le problème du plus court chemin sans contraintes de ressources utilise la programmation dynamique. On distingue parmi ces

méthodes, celles qui sont une extension de l'algorithme de Ford-Bellman (qui fonctionnent par correction d'étiquettes) et celles qui sont une extension de l'algorithme de Dijkstra (qui utilisent un marquage permanent des étiquettes lorsque les longueurs des arcs sont non négatives). Nous commencerons par présenter ces deux classes d'algorithmes.

Algorithme de marquage d'étiquettes

Soit un graphe $G = (N, A)$ tel que $|N| = n$. L'algorithme effectue des tentatives d'étiquetage aux sommets dans l'espoir de trouver la distance minimale du noeud origine o à tous les noeuds du graphe (i.e., une étiquette permanente). Il maintient pour cela deux ensembles de noeuds. L'ensemble $PERM$ qui contient les noeuds auxquels on a pu associer une étiquette permanente, et l'ensemble $TENT$ qui contient les noeuds non encore marqués de façon permanente. Pour tout noeud $j \in A$, soit $d(j)$ la distance du plus court chemin du noeud origine o au noeud j . On a alors l'algorithme suivant :

```

PERM ← ∅
TENT ← N
d(i) ← ∞   ∀i ∈ N,   d(o) ← 0
Tant que |PERM| < n faire {
    choisir i ∈ TENT avec d(i) = min{d(j) : j ∈ TENT}
    PERM ← PERM ∪ {i}
    TENT ← TENT \ {i}
    Pour tout (i, j) ∈ A faire
        si d(j) > d(i) + cij alors d(j) ← d(i) + cij
}

```

Cet algorithme est connu sous le nom de l'algorithme de Dijkstra et a une complexité de $\Theta(n^2)$ pour une implémentation naïve. Cependant, plusieurs implémentations plus sophistiquées de cet algorithme ont été proposées pour améliorer cette complexité.

Algorithme de correction d'étiquettes

L'algorithme par correction d'étiquettes maintient une étiquette sur la distance courante $d(j)$ menant à chaque noeud j . Celles-ci sont mises à jour itérativement jusqu'à ce que la distance minimale du noeud origine o à chaque noeud j du graphe soit trouvée. L'algorithme s'arrête lorsque tous les $d(j)$ satisfont la condition d'optimalité $d(j) \leq d(i) + c_{ij}$, $\forall (i, j) \in A$. L'ensemble *LIST* contient la liste des éléments à traiter. Tant qu'il y a encore des sommets à traiter ($LIST \neq \emptyset$), l'algorithme choisit un élément i et met à jour les $d(j)$ des sommets adjacents à i (s'il y a lieu). S'il y a eu mise à jour, alors ce sommet sera placé dans *LIST* pour être traité ultérieurement.

Notons que l'algorithme fonctionne pour des coûts quelconques, mais le graphe ne doit pas contenir de circuit de coût négatif.

$$d(i) \leftarrow \infty, \quad \forall i \in N, \quad d(o) \leftarrow 0$$

$$LIST \leftarrow \{o\}$$

Tant que $LIST \neq \emptyset$ faire {

choisir $i \in LIST$ et $LIST \leftarrow LIST \setminus \{i\}$

Pour tout $(i, j) \in A$ faire

si $d(j) > d(i) + c_{ij}$ alors {

$d(j) \leftarrow d(i) + c_{ij}$

$$\begin{aligned}
 & \text{si } j \notin LIST \text{ alors } LIST \leftarrow LIST \cup \{j\} \\
 & \quad } \\
 & \quad } \\
 & \quad }
 \end{aligned}$$

Cet algorithme est connu sous le nom de l'algorithme de Ford-Bellman et a une complexité de $\Theta(nm)$.

3.4 Plus Courts Chemins avec Contraintes de Ressources

3.4.1 Approches par étiquetage

Dans les algorithmes de recherche du plus court chemin sans contraintes, on maintient pour chaque noeud j la distance du plus court chemin du noeud origine o au noeud j . Dans le cas avec contraintes, une étiquette consiste en un couple (C_q, T_q) qui représente respectivement le coût et le vecteur de consommation des ressources tout au long du chemin q menant de o à j . On doit maintenir une liste d'étiquettes par noeud et il n'existe plus nécessairement d'ordre total entre les noeuds.

Les approches par étiquetage peuvent être vues comme une amélioration des méthodes de programmation dynamique dans le sens où elles ne considèrent que les chemins non-dominés. Seuls les chemins non-dominés sont stockés en ordre croissant de leurs coûts dans la liste des étiquettes associées à chaque noeud (soit dans l'ordre décroissant des ressources, dans le cas d'une seule ressource). En commençant avec une liste d'étiquettes vide à chaque sommet, exceptée l'étiquette $(0, \dots, 0)$ au noeud o , l'algorithme étend la

liste des étiquettes en prolongeant les sous-chemins non dominés aux noeuds adjacents. Par ailleurs, un chemin n'est pas étendu si la consommation d'une ressource excède une limite donnée.

Contraintes de fenêtres de temps

La consommation de la ressource temps dans le cas de problèmes de plus court chemin avec fenêtres de temps aux sommets, se fait de la façon suivante. Soit P_i un chemin du noeud o au noeud i . Supposons que le temps d'arrivée au noeud i est t_i et que la fenêtre de temps au noeud i est $[a_i, b_i]$. Si $t_i > b_i$, alors le chemin P_i est éliminé. Sinon, la consommation de temps sur l'arc (i, j) est $t_{ij} + \max\{a_i - t_i, 0\}$.

Contraintes de délai de service

Chaque client i doit être servi avant un délai maximal t_{max} . Soit r une route partielle du noeud o au noeud i . Soit \underline{t}_i^r le temps d'arrivée au plus tôt au noeud i en partant du noeud o au temps 0. Si $(\underline{t}_i^r - 0) \leq t_{max}$ alors on accepte le chemin P_i . Autrement, soient \bar{t}_i^r le temps d'arrivée au plus tard au noeud i et \bar{t}_0^r le temps de départ au plus tard de la tournée. Si $(\bar{t}_i^r - \bar{t}_0^r) \leq t_{max}$ alors le chemin P_i est accepté et la consommation de temps sur l'arc (i, j) est $t_{ij} + \max\{a_i - \bar{t}_i^r, 0\}$. Sinon, le chemin P_i est éliminé.

Contraintes de capacité

Soit Q la capacité du véhicule, Q_i la quantité accumulée sur le chemin P_i , et q_i la quantité demandée par le client i . Si $Q_i + q_i > Q$ alors le chemin P_i est éliminé. Sinon la consommation de la ressource capacité est de $Q_i + q_i$.

Contraintes de chemin élémentaire

Une autre contrainte impose que le chemin trouvé soit élémentaire (i.e., qu'il ne contienne pas de cycles). En un sens, on voit ici chaque sommet comme une ressource. Si le graphe contient des circuits de coût négatif, il est évident que le chemin le plus court tendra à parcourir ces circuits indéfiniment pour réduire le coût. L'approche de Feillet et al. (2000) permet justement d'éviter cela et de trouver des chemins élémentaires.

3.4.2 Approche de Feillet et al.

Feillet et al. (2000) ont proposé un algorithme pour la résolution du problème du plus court chemin élémentaire avec contraintes de ressources. Dans ce type de problème avec plusieurs objectifs, on s'intéresse aux chemins non dominés. L'algorithme proposé est capable de résoudre le problème du plus court chemin élémentaire dans n'importe quel type de graphes, même dans des graphes qui comportent des circuits de coût négatif. L'objectif est donc de construire tous les chemins élémentaires non dominés partant de o et allant jusqu'à un certain sommet j et satisfaisant toutes les contraintes de ressources.

L'algorithme maintient pour chaque chemin partiel allant de o à j , une étiquette qui contient, la longueur du chemin, le vecteur des quantités de ressources consommées par le chemin, et la trace des sommets visités.

S'il y a deux chemins différents menant de o à j , alors l'un domine l'autre si toutes les conditions suivantes sont vérifiées :

- (1) il est de coût plus faible,
- (2) il consomme moins de ressources, pour chaque ressource considérée,
- (3) l'ensemble des sommets inaccessibles, soit parce qu'ils ont déjà été visités, soit parce qu'ils demanderaient une consommation de ressource excédant les disponibilités, est un sous-ensemble de l'ensemble des sommets inaccessibles de l'autre chemin.

Dans la suite, nous décrivons plus formellement cette approche.

Notation

Supposons que le nombre de ressources est égal à L . A chaque chemin X_{oj} d'origine o et d'extrémité j est associé un état $R_j = (T_j^1, \dots, T_j^L, s_j, C(T_j^1, T_j^2, \dots, T_j^L), V_j^1, \dots, V_j^n)$ correspondant à la quantité consommée de chacune des L ressources par le chemin, au coût $C(T_j^1, T_j^2, \dots, T_j^L)$, au nombre s_j de noeuds inaccessibles, et à la trace des sommets inaccessibles ($V_j^i = 1$ si ce sommet est inaccessible, 0 sinon). Feillet et al. (2000) ont introduit le critère de dominance suivant :

Définition

Soient X'_{oj} and X^*_{oj} deux chemins distincts allant de o à j et leurs étiquettes associées (R'_j, C'_j) et (R^*_j, C^*_j) . Alors, X'_{oj} domine X^*_{oj} si et seulement si $C'_j \leq C^*_j$, $s'_j \leq s_j$, $T_j^{l'} \leq T_j^{*l}$ pour $l = 1, \dots, L$, $V_j^{l'} \leq V_j^{*l}$ pour $l = 1, \dots, n$ et $(R'_j, C'_j) \neq (R^*_j, C^*_j)$.

Description de l'algorithme

Nous utilisons la notation suivante :

- L_i : Liste des étiquettes au noeud i .
- $Succ(i)$: Liste des successeurs du noeud i .
- E : Liste des noeuds en attente d'être traités.
- $Prolonge(l_i, j)$: Fonction qui retourne l'étiquette résultant de l'extension de l'étiquette l_i au noeud j . Elle met à jour la consommation des différentes ressources et vérifie que les contraintes sont respectées.
- F_{ij} : Liste des étiquettes au noeud j étendues à partir du noeud i .
- $Insertion(F_{ij}, L_j)$: Procédure qui insère les étiquettes F_{ij} dans L_j et conserve seulement les étiquettes non-dominées dans la liste des étiquettes L_j .

Alors, l'algorithme est le suivant :

Algorithme

```

1.  $L_o \leftarrow (0, \dots, 0), L_i \leftarrow \emptyset \quad \forall i \in V \setminus \{o\}, E \leftarrow \{o\}$ 
2. Tant que  $E \neq \emptyset$  {
  % Sélection du prochain sommet à traiter
  3. Choisir  $i \in E$ 
  % Traitement des étiquettes au noeud  $i$ 
  4. Pour tout  $j \in \text{succ}[i]$  faire {
    4.1.  $F_{ij} \leftarrow \emptyset$ 
    4.2. Pour tout  $l_i \in L_i$  faire
      4.3. Si  $V_i^j = 0$  alors {
        4.3.1.  $F_{ij} \leftarrow F_{ij} \cup \{\text{Prolonge}(l_i, j)\}$ 
        4.3.2. Si  $L_j \neq \text{Insertion}(F_{ij}, l_j)$  alors {
          4.3.3.  $L_j \leftarrow \text{Insertion}(F_{ij}, l_j)$ 
          4.3.4.  $E \leftarrow E \cup \{j\}$ 
        }
      }
    }
  }
  % Réduction de  $E$ 
  5.  $E \leftarrow E \setminus \{i\}$ 
}

```

L'initialisation se fait avec l'étiquette $(0, \dots, 0)$ associée au noeud origine o (Étape 1).

L'étape 3 consiste à choisir un élément de la liste E à traiter. Le traitement d'une étiquette

(Etape 4) au noeud i consiste à mettre à jour les étiquettes de chacun des noeuds successeurs qui sont accessibles (au sens de la définition donnée précédemment). Cette extension se fait par la fonction *prolonge* (Etape 4.3.1). Si la nouvelle étiquette n'est pas dominée par une étiquette déjà existante au noeud j , alors elle sera rajoutée à la liste des étiquettes du noeud j . Par ailleurs, toutes les étiquettes déjà existantes au noeud j sont supprimées si elles sont dominées par cette nouvelle étiquette (Etape 4.3.3). Les étiquettes correspondant au sommet $n + 1$ (retour au dépôt) ne sont plus étendues et sont gardées dans l'ensemble des solutions. Enfin, lorsqu'un sommet est traité, il est supprimé de la liste des sommets à traiter (Etape 5).

Il est possible d'implanter cet algorithme de manière efficace pour réduire le nombre d'opérations à réaliser. En pratique, la complexité de l'algorithme est fortement liée à la structure du graphe et aux contraintes de ressources.

Chapitre 4

Méthode de résolution

Nous proposons ici une méthode de résolution pour notre problème de livraison avec fenêtres de temps et utilisation multiple de véhicules pour le cas d'un seul véhicule. Il s'agit d'une méthode exacte basée sur une modélisation du problème sous forme de graphe dans lequel on recherche des plus courts chemins sous différentes contraintes telles la capacité, le temps, et le délai de service. Puisque le revenu associé à chaque client est non renouvelable et ne peut être exploité qu'une seule fois, on n'a aucun intérêt à revenir à un client déjà visité. On recherche donc des chemins plus courts élémentaires. On s'intéresse ici au cas où le revenu associé à chaque client i vaut 1 (i.e., $p_i = 1$) et où la constante α' est suffisamment grande.

Notre méthode de résolution exploite l'algorithme de recherche des plus courts chemins élémentaires avec contraintes de ressources de Feillet et al. (2000) et se divise en deux phases. Celles-ci sont décrites dans la suite.

4.1 Phase 1 - Génération des tournées avec contraintes de ressources

Pour générer les tournées réalisables, on utilise l'algorithme des plus courts chemins élémentaires avec contraintes de ressources de Feillet et al. (2000). Les ressources dans notre cas sont la capacité, les fenêtres de temps associées aux clients, et le délai de service aux clients. Cet algorithme est exploité afin de générer toutes les tournées réalisables, mais en évitant une énumération systématique de toutes les possibilités. Plus particulièrement, si deux tournées partielles visitent les mêmes sommets (mais pas dans le même ordre) et que l'une des tournées est de coût supérieur et consomme plus de ressources, alors elle est dominée et peut être éliminée.

Une énumération complète est possible à cause de la contrainte de délai de service qui limite beaucoup le nombre de clients pouvant être desservis dans chaque tournée. Pour illustrer ceci, si l'on considère une instance à 10 clients, il existe $C_i^{10}i!$ tournées de i clients, i.e., $\sum_{i=1,\dots,10} C_i^{10}i! = 6,235,200$ tournées. Si l'on considère que la contrainte de délai de service est telle que l'on ne peut desservir que 3 clients au maximum par tournée, alors ce nombre n'est plus que de $\sum_{i=1,\dots,3} C_i^{10}i! = 820$ tournées.

Avant d'appliquer l'algorithme de Feillet et al., les distances $d_{i,j}$ sont remplacées par $d_{i,j} - \alpha$ sur chaque arc du graphe, avec $\alpha > \max_{(i,j) \in A} d_{i,j}$. Cette modification génère des arcs de coût négatif, et force l'algorithme à visiter des clients (autrement, la solution optimale serait de rester au dépôt). De plus, le critère de dominance est modifié comme suit :

Soient X'_{oj} and X^*_{oj} deux chemins distincts allant de o à j et leurs étiquettes associées (R'_j, C'_j) et (R^*_j, C^*_j) . Alors, X'_{oj} domine X^*_{oj} si et seulement si $C'_j \leq C^*_j$, $T_j^{l'} \leq T_j^{*l}$ pour $l = 1, \dots, L$, $V_j^{l'} = V_j^{*l}$ pour $l = 1, \dots, n$ et $(R'_j, C'_j) \neq (R^*_j, C^*_j)$.

Notons que dans cette nouvelle définition de dominance, le vecteur V nous indique seulement quels sont les clients qui ont été visités dans la tournée partielle (i.e., $V_i^l = 1$ si le sommet l est visité, 0 sinon).

Les figures 4.1 et 4.2 montrent deux tournées partielles r et r' qui visitent les mêmes sommets pour se rendre de o à j mais dans un ordre différent. Si la tournée r domine r' au sens de la définition précédente, il ne sert à rien d'étendre la tournée r' avec de nouveaux sommets et on peut donc l'éliminer. En général, le critère de dominance nous permet donc de réduire l'espace de recherche en ne conservant que les tournées non-dominées.

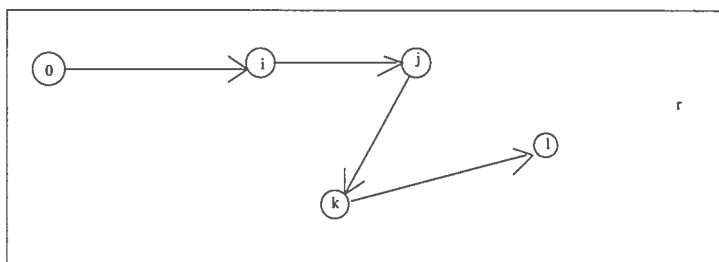
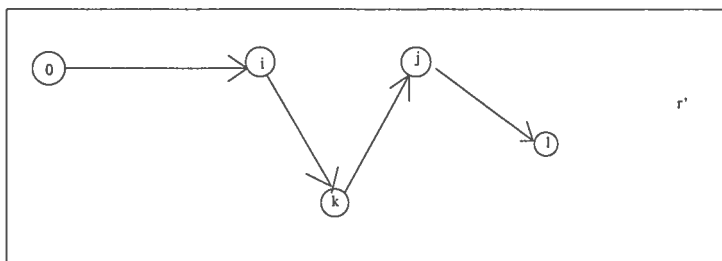


Figure 4.1 – Une tournée partielle r

Figure 4.2 – Une tournée partielle r'

4.2 Phase 2 - Construction du graphe des tournées

Afin de constituer une journée de travail pour le véhicule, on va sélectionner et séquencer des tournées générées au cours de la phase 1, en appliquant le même algorithme de chemins plus courts à un graphe de tournées. Dans ce graphe, les noeuds sont les tournées générées à la fin de la phase 1, plus deux noeuds artificiels pour représenter le début et la fin de la journée. Un arc existe entre deux tournées r et r' s'il est possible d'effectuer la tournée r' après la tournée r . Le coût associé à l'arc (r, r') est la longueur totale de la tournée r' auquel on soustrait α' fois le nombre de clients de la tournée r' , soit

$$\sum_{(i,j) \in A} d_{ij} x_{ij}^{r'} - \alpha' \sum_{i \in N} y_i^{r'}$$

$$\text{avec } \alpha' > 2 \max_{i \in N} \{ \max \{ d_{0i}, d_{i,n+1} \} \}$$

Ainsi les coûts de tous les arcs sont négatifs. En effet, si on considère les clients visités dans la tournée r' , et en supposant que l'inégalité triangulaire est vérifiée, nous avons :

$$\sum_{(i,j) \in A} d_{ij} x_{ij}^{r'} \quad (4.1)$$

$$\leq \sum_{i \in N} (d_{oi} + d_{i,n+1}) y_i^{r'} \quad (4.2)$$

$$\leq \sum_{i \in N} 2 \max\{d_{oi}, d_{i,n+1}\} y_i^{r'} \quad (4.3)$$

$$\leq 2 \max_{i \in N} \{\max\{d_{oi}, d_{i,n+1}\}\} \sum_{i \in N} y_i^{r'} \quad (4.4)$$

$$< \alpha' \sum_{i \in N} y_i^{r'} \quad (4.5)$$

Le passage de l'équation (4.1) à l'équation (4.2) vient du fait que le service à chaque client par une tournée individuelle est plus coûteux que toute tournée desservant l'ensemble des clients (à cause de l'inégalité triangulaire).

On retrouve également deux fenêtres de temps associées au début et à la fin de chaque tournée. Celles-ci sont décrites dans la suite.

4.2.1 Temps de début et de fin au plus tard d'une tournée

Pour déterminer le temps de début au plus tard d'une tournée, on effectue un balayage à rebours des clients de la tournée en partant de la borne supérieure de la fenêtre de temps au dépôt. Supposons que la tournée r correspond à la séquence $(0 = i_0, i_1, i_2, \dots, i_{n_r}, i_{n_r+1} = n+1)$, où n_r est le nombre de clients de la tournée. Nous déterminons d'abord le temps de début de service à chaque client de la tournée $\bar{t}_{i_j}^r$ (où $\bar{t}_{i_0}^r = \bar{t}_0^r$ et $\bar{t}_{i_{n_r+1}}^r = \bar{t}_{n+1}^r$ sont les temps de départ et d'arrivée au plus tard au dépôt, respectivement). Un balayage arrière de la tournée r est appliqué à partir de i_{n_r+1} jusqu'à i_0 comme suit :

$$\begin{aligned}\bar{t}_{i_{n_r+1}}^r &\leftarrow b_{i_{n_r+1}}, \\ \bar{t}_{i_j}^r &\leftarrow \min\{\bar{t}_{i_{j+1}}^r - t_{i_j i_{j+1}} - s_{i_j}, b_{i_j}\}, j = i_{n_r}, \dots, i_0 = 0.\end{aligned}$$

Une fois que $\bar{t}_{i_0}^r$ est obtenu, un balayage avant est effectué pour mettre à jour les $\bar{t}_{i_j}^r$ et obtenir l'horaire au plus tard de cette tournée.

$$\bar{t}_{i_j}^r \leftarrow \max\{\bar{t}_{i_{j-1}}^r + s_{i_{j-1}} + t_{i_{j-1} i_j}, a_{i_j}\}, j = i_1, \dots, i_{n_r+1} = n + 1.$$

La durée de la tournée correspond à $d_{min}^r = \bar{t}_{i_{n_r+1}}^r - \bar{t}_0^r$. Cette durée est minimale car en partant au plus tard, on minimise les temps d'attente aux clients.

4.2.2 Temps de début et de fin au plus tôt d'une tournée

Nous voulons maintenant déterminer les temps de départ et d'arrivée au plus tôt au dépôt (avec $\underline{t}_{i_0}^r = \underline{t}_0^r$ et $\underline{t}_{i_{n_r+1}}^r = \underline{t}_{n+1}^r$), tout en maintenant la durée minimale de la tournée d_{min}^r . Deux cas sont à considérer :

Cas 1. S'il n'y a pas d'attente dans l'horaire au plus tard, alors on peut avancer les temps de début de service au plus tard de δ^r unités de temps, où :

$$\delta^r = \min_{j=0, \dots, n_r+1} \bar{t}_{i_j}^r - a_{i_j}.$$

Les fenêtres de temps pour le début et la fin de la tournée sont donc : $[\underline{t}_0^r; \bar{t}_0^r]$ et $[\underline{t}_{n+1}^r; \bar{t}_{n+1}^r]$, où $\underline{t}_0^r = \bar{t}_0^r - \delta^r$ et $\underline{t}_{n+1}^r = \bar{t}_{n+1}^r - \delta^r$.

Cas 2. S'il y a attente dans l'horaire au plus tard, alors il n'est pas possible de partir du dépôt plus tôt sans allonger la durée de la tournée. Par conséquent, nous avons $\underline{t}_0^r = \bar{t}_0^r$ et $\underline{t}_{n+1}^r = \bar{t}_{n+1}^r$. Dans ce cas, les fenêtres de temps pour le début et la fin de la tournée se réduisent à un seul point.

Le calcul de ces intervalles de temps nous permet de déterminer les arcs valides dans le graphe des tournées. En effet, un arc existe entre les noeuds (tournées) r et r' dans ce graphe, si :

$$\underline{t}_{n+1}^r + \sigma^{r'} \leq \bar{t}_0^{r'}.$$

Un fois le graphe des tournées construit, on cherche un plus court chemin élémentaire avec une seule contrainte de ressource pour les fenêtres de temps associées aux tournées. On utilise pour cela l'algorithme de Feillet et al. (2000) avec une contrainte de ressource qui est le temps.

Exemple

Nous allons illustrer ces calculs par un exemple. Prenons un dépôt et les 5 clients suivants avec leurs fenêtres de temps correspondantes :

0 - $[0, \infty]$.

1 - $[591, 874]$

2 - $[73, 350]$

3 - $[473, 588]$

4 - [418 , 913]

5 - [40 , 390]

Considérons la tournée suivante d'indice 1 : 0 - 3 - 1 - 6.

Les distances sont les suivantes : $d_{03} = 39.35$, $d_{31} = 3$, $d_{10} = 38.07$.

Les durées de service à chaque client : 10 .

$$\bar{t}_6^1 = \infty.$$

$$\bar{t}_1^1 = \min\{\bar{t}_6^1 - 38.07 - 10, 874\} = 874.$$

$$\bar{t}_3^1 = \min\{\bar{t}_1^1 - 3 - 10, 588\} = 588.$$

$$\bar{t}_0^1 = \min\{\bar{t}_3^1 - 39.35 - 0, \infty\} = 548.65.$$

D'où $\bar{t}_0^1 = 548.65$.

Maintenant :

$$\bar{t}_0^1 = 548.65.$$

$$\bar{t}_3^1 = \max\{\bar{t}_0^1 + 39.35 + 0, 473\} = 588.$$

$$\bar{t}_1^1 = \max\{\bar{t}_3^1 + 3 + 10, 591\} = 601.$$

$$\bar{t}_6^1 = \max\{\bar{t}_1^1 + 38.07 + 10, 0\} = 649.07.$$

Durée minimale du trajet = $649.07 - 548.65 = 100.42$.

Calculons maintenant

$$\delta^1 = \min_{j=0,\dots,n+1} \bar{t}_{i_j}^1 - a_{i_j}.$$

$$\min\{548.65 - 0, 588 - 473, 601 - 591, 649.07 - 0\} = 10$$

Les fenêtres de temps pour le début et la fin de la tournée 1 sont : $[\underline{t}_0^1, \bar{t}_0^1] = [538.65, 548.05]$

et $[\underline{t}_{n+1}^1, \bar{t}_{n+1}^1] = [639.07, 649.07]$, où $\underline{t}_0^1 = \bar{t}_0^1 - \delta^1 = 548.65 - 10 = 538.65$ et $\underline{t}_{n+1}^1 = \bar{t}_{n+1}^1 - \delta^1 = 649.07 - 10 = 639.07$.

Nous présentons les fenêtres pour toutes les tournées générées suite à la phase 1 dans le tableau 4.1. Pour chaque tournée, on donne respectivement la séquence des clients desser-

vis, les temps de début et de fin au plus tôt, les temps de début et de fin au plus tard et le coût de la tournée. Notons que t_{max} a été fixé à 55, ce qui signifie que le délai de service pour chaque client de chaque tournée ne doit pas dépasser cette valeur.

Tournées	Début au Plus tôt	Début au Plus tard	Fin au Plus tôt	Fin au Plus tard	Coût
- 0 3 1 0 -	538.65	548.65	639.07	649.07	-82.82
- 0 4 3 0 -	421.57	536.57	522.35	637.35	-82.46
- 0 4 1 0 -	537.88	820.88	639.07	922.07	-82.05
- 0 2 5 0 -	42.2	319.2	143.5	420.5	-81.94
- 0 2 0 -	42.2	319.2	113.8	390.8	-20.02
- 0 4 0 -	381.95	876.95	464.05	959.05	-9.52
- 0 1 0 -	552.93	835.93	639.07	922.07	-5.48
- 0 3 0 -	433.65	548.65	522.35	637.35	-2.92
- 0 5 0 -	0	349.69	90.62	440.31	-1

Tableau 4.1 – Fenêtres de temps associées aux tournées.

Le graphe des tournées est présenté à la figure 4.2. Remarquons qu'on a omis de représenter tous les arcs aboutissant au sommet de fin de journée (qui sont tous de coût nul), et ceux originant du sommet de début de journée (qui ont des coûts qui correspondent aux coûts des tournées destinations). Seuls les arcs appartenant au chemin plus court élémentaire sont représentés. Ce chemin est représenté en ligne discontinue.

La journée optimale associée au véhicule est donc la séquence de tournées 0 2 5 0 - 0 4 0 - 0 3 1 0. Dans ce cas, le véhicule est en mesure de revenir au dépôt à la fin de sa journée au temps le plus tôt soit 639.07.

Nous venons de présenter l'approche de résolution que nous avons mise en œuvre pour

notre problème de livraison avec réutilisation de véhicules. Nous présenterons par la suite les expérimentations menées afin d'évaluer notre algorithme sur les instances de Solomon (1987).

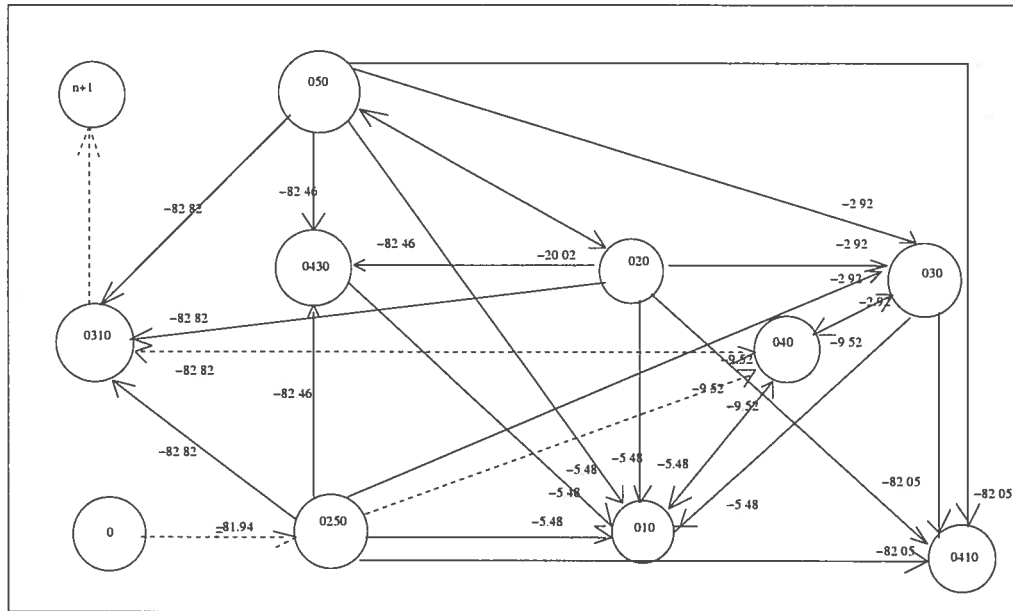


Figure 4.3 – Graphe de tournées

Chapitre 5

Expérimentation

Nous avons testé notre approche sur les problèmes de Solomon (1987). Tous ces problèmes ont 100 clients, répartis dans un espace euclidien. Les temps de parcours correspondent aux distances euclidiennes. Il y a six classes de problèmes qui se différencient d'abord par la distribution géographique des clients. Dans les instances C (Clustered), les clients sont regroupés géographiquement. Dans les instances R (Random), les clients ont des positions géographiques aléatoires. Dans les instances RC (Random Clustered), les clients ont des positions géographiques mixtes (aléatoires et groupés).

Les problèmes de type 1 (R1, C1 et RC1) ont un horizon de planification court permettant de desservir peu de clients par tournée (5 à 10 clients approximativement). Au contraire, les problèmes de type 2 (R2, C2 et RC2) ont un horizon plus long et permettent de desservir plus de clients par véhicule. Les coordonnées des clients sont identiques pour chaque classe. Les problèmes diffèrent au niveau de la largeur des fenêtres de temps. Pour certains, les fenêtres de temps sont larges. Pour d'autres, elles sont plus contraignantes.

Nos tests ont porté sur les problèmes de type 2 où l'horizon est plus large et permet donc d'enchaîner plusieurs tournées. Ces tests sont divisés en deux parties. Nous avons commencé par mener une expérimentation montrant qu'il est possible d'utiliser l'algorithme

décrit dans la section 3.4.2 pour résoudre notre problème et évaluer le nombre de clients que l'on peut desservir avec un véhicule au fur et à mesure que le délai de service t_{max} augmente. Ces tests ont été effectués avec les instances classiques de Solomon. Ensuite, des tests ont été réalisés en considérant des ensembles de clients à desservir plus restreints, en l'occurrence ceux qu'on retrouve sur des tournées individuelles appartenant aux meilleures solutions rapportées dans la littérature sur les mêmes instances de Solomon.

Lors de nos expérimentations, nous avons supposé une capacité infinie pour le véhicule. En effet, cette contrainte n'est pas liante par rapport aux contraintes de fenêtres de temps et de délai de service (c'est, en fait, cette dernière contrainte qui limite le nombre de clients par tournée). Les durées et coûts de parcours entre deux sommets ont été tronqués à deux décimales. De plus, la constante β dans l'équation (2.13) du modèle mathématique décrit à la section 2.1.3 a été fixée à 1/5.

Tous les tests ont été réalisés sur une machine SUN ULTRA III cadencée à 900 MHz possédant 8G de RAM et fonctionnant sous UNIX. Les programmes ont été écrits en C++ en utilisant le compilateur CC version 5.3.

5.1 Effet du critère de dominance en phase 1

La modification que nous avons apportée au critère de dominance présenté à la section 4.2, nous a permis de réduire de manière significative le nombre de tournées que l'on doit considérer, et de générer toutes les tournées réalisables et non dominées. Nous avons relâché ce critère pour générer toutes les routes possibles. Les résultats sont donnés dans le tableau 5.1. On donne pour chaque classe de problème la moyenne du nombre de tournées

généérées avec et sans le critère de dominance. La dernière colonne donne le gain obtenu en pourcentage. Ce gain est en moyenne de 22.98% et varie plus précisément entre 19.26% et 28.38% selon la classe de problèmes.

Type Instance	Sans critère	Avec critère	Gain en %
R	30695.81	21984.12	28.38%
RC	4384.33	3450.45	21.30%
C	2145.58	1732.25	19.26%

Tableau 5.1 – Moyenne du nombre de tournées générées en phase 1.

5.2 Résultats sur les instances de Solomon de type 2

Nous allons présenter les résultats que nous avons obtenus dans les tableaux 5.2 et suivants, en considérant les 25, 50, et 100 premiers clients des instances de Solomon pour deux valeurs différentes de t_{max} . Pour chaque problème, nous donnons le nombre de tournées réalisables générées à la fin de la phase 1, le nombre de tournées qui ont été sélectionnées dans la journée de travail, la distance totale parcourue par le véhicule, le nombre de clients desservis par le véhicule, le nombre moyen de clients par tournée, et le temps d'exécution en secondes.

Les solutions que nous présentons sont optimales par rapport à l'objectif hiérarchique que nous nous sommes fixés, à savoir maximiser le nombre de clients à desservir comme premier critère et minimiser la distance totale parcourue comme second critère. Les cases non remplies dans les tableaux correspondent aux instances qu'on n'a pu résoudre étant donné

le grand nombre de routes générées en phase 1. Il est important de remarquer qu'à cause de la composante aléatoire des problèmes R et RC, ces problèmes se sont révélés plus difficiles à résoudre.

Prob	# Tournées Phase 1	# Tournées Phase 2	Distance parcourue	# Clients desservis	% Clients desservis	# Moyen de clients	Temps d'exécution
r201.25	122	8	315.32	19	76%	2.37	1
r201.50	550	9	440.20	25	50%	2.77	20
r201.100	4279	11	379.79	36	36%	3.27	16156
r202.25	341	9	337.55	20	80%	2.22	3
r202.50	2223	10	430.63	28	56%	2.8	1075
r202.100	32495						
r203.25	443	7	376.44	20	80%	2.85	18
r203.50	3248	9	434.85	30	60%	3.33	8081
r203.100	62685						
r204.25	547	8	407.77	23	92%	2.87	45
r204.50	4535	9	448.50	32	64%	3.55	33068
r204.100	87542						
r205.25	237	9	433.98	23	92%	2.55	4
r205.50	1424	9	457.12	29	57%	3.22	596
r205.100	16399						
r206.25	425	9	416.15	22	88%	2.44	11
r206.50	3059	10	463.30	30	60%	3	4078
r206.100	50575						
r207.25	480	9	441.85	24	96%	2.66	20
r207.50	3682	10	422.04	32	64%	3.2	11326
r207.100	73008						
r208.25	552	8	420.02	24	96%	3	44
r208.50	4548	9	430.76	32	64%	3.55	34041
r208.100	93918						
r209.25	381	9	465.98	23	92%	2.55	11
r209.50	2531	8	421.92	28	56%	3.5	2754
r209.100	42731						
r210.25	397	8	429.35	21	84%	2.62	11
r210.50	3160	9	440.26	29	57%	3.22	5051
r210.100	51537						
r211.25	597	9	412.72	24	96%	2.66	41
r211.50	5067	9	460.55	30	60%	3.33	19588
r211.100	124008						

Tableau 5.2 – Solutions pour les instances R avec $t_{max} = 75$.

Prob	# Tournées Phase 1	# Tournées Phase 2	Distance parcourue	# Clients desservis	% Clients desservis	# Moyen de clients	Temps d'exécution
r201.25	217	8	392.49	20	80%	2.5	1
r201.50	1263	9	419.71	26	52%	2.88	203
r201.100	13729	10	404.44	37	37%	3.7	245906
r202.25	778	7	304.48	20	80%	2.85	26
r202.50	6800	8	452.80	29	57%	3.62	16410
r202.100							
r203.25	1097	6	397.08	21	84%	3.5	98
r203.50	11320	9	443.13	33	66%	3.66	141630
r203.100							
r204.25	1435	6	361.30	22	88%	3.66	385
r204.50	7209	8	476.10	33	66%	4.12	82361
r204.100							
r205.25	527	7	351.62	22	88%	3.14	21
r205.50	4003	8	450.91	29	57%	3.62	8441
r205.100							
r206.25	1034	8	459.55	24	96%	3	89
r206.50	10491	8	482.19	32	64%	4	100630
r207.100							
r207.25	1236	7	413.03	23	92%	3.28	221
r207.50	13539	8	417.57	33	66%	4.12	316096
r207.100							
r208.25	1462	6	384.59	23	92%	3.83	534
r208.txt	7237	8	431.19	33	66%	4.12	89596
r208.100							
r209.25	902	8	425.62	23	92%	2.87	64
r209.50	8245	9	441.54	30	60%	3.33	46973
r209.100							
r210.25	986	8	441.38	22	88%	2.75	67
r210.50	11144	8	450.36	30	60%	3.75	113942
r210.100							
r211.25	1649	8	426.12	25	100%	3.12	422
r211.50	8123	9	461.67	31	62%	3.44	60820
r211.100							

Tableau 5.3 – Solutions pour les instances R avec $t_{max} = 90$.

Prob	# Tours Phase 1	# Tours Phase 2	Distance parcourue	# Clients desservis	% Clients desservis	# Moyen de clients	Temps d'exécution
rc201.25	74	8	339.09	16	64%	2	1
rc201.50	174	7	340.15	19	38%	2.71	1
rc201.100	1562	10	426.16	28	28%	2.8	877
rc202.25	167	7	328.88	18	72%	2.57	1
rc202.50	395	7	358.04	22	44%	3.14	10
rc202.100	5210	10	439.27	32	32%	3.2	24561
rc203.25	252	7	327.47	20	80%	2.85	3
rc203.50	587	7	342.13	24	48%	3.42	47
rc203.100	9877	8	398.72	33	33%	4.12	188029
rc204.25	320	8	407.29	21	84%	2.62	7
rc204.50	866	7	325.32	24	48%	3.42	160
rc204.100	9141	9	396.14	34	34%	3.77	202461
rc205.25	147	7	348.16	18	72%	2.57	1
rc205.50	353	7	353.75	22	44%	3.14	6
rc205.100	4036	9	383.28	30	30%	3.33	9389
rc206.25	138	7	327.44	18	72%	2.57	1
rc206.50	334	8	381.49	23	46%	2.87	6
rc206.100	4966	9	366.41	31	31%	3.44	23338
rc207.25	258	8	385.42	20	80%	2.5	2
rc207.50	680	7	366.82	24	48%	3.42	40
rc207.100	11016	9	420.9	32	32%	3.55	278804
rc208.25	386	8	370.71	22	88%	2.75	10
rc208.50	998	8	364.35	25	50%	3.12	196
rc208.100	10640	9	379.36	32	32%	3.55	188745

Tableau 5.4 – Solutions pour les instances RC avec $t_{max} = 75$.

Prob	# Tournées Phase 1	# Tournées Phase 2	Distance parcourue	# Clients desservis	% Clients desservis	# Moyen de clients	Temps d'exécution
rc201.25	132	8	342.88	17	68%	2.12	0
rc201.50	312	6	328.97	21	42%	3.5	5
rc201.100	4456	10	393.91	29	29%	2.9	15449
rc202.25	323	8	356.30	20	80%	2.5	4
rc202.50	911	7	378.45	26	52%	3.71	74
rc202.100	20000						
rc203.25	493	7	332.04	22	88%	3.14	13
rc203.50	1413	7	347.17	28	56%	4	723
rc203.100	45811						
rc204.25	708	7	350.02	23	92%	3.28	64
rc204.50	2404	7	348.14	30	60%	4.28	5175
rc204.100	89808						
rc205.25	264	7	359.82	20	80%	2.85	2
rc205.50	753	7	364.71	24	48%	3.42	54
rc205.100	15161						
rc206.25	259	6	324.69	20	80%	3.33	5
rc206.50	727	7	370.13	25	50%	3.57	73
rc206.100	18989						
rc207.25	560	7	349.52	21	84%	3	18
rc207.50	1792	7	359.31	25	50%	3.57	844
rc207.100	53233						
rc208.25	954	7	384.93	23	92%	3.28	136
rc208.50	3117	7	363.96	27	54%	3.85	4800
rc208.100	115098						

Tableau 5.5 – Solutions pour les instances RC avec $t_{max} = 90$.

Prob	# Tournées Phase 1	# Tournées Phase 2	Distance parcourue	# Clients desservis	% Clients desservis	# Moyen de clients	Temps d'exécution
c201.25	102	10	326.44	22	88%	2.2	1
c201.50	340	12	356.62	28	56%	2.33	10
c201.100	1204	11	320.85	30	30%	2.72	547
c202.25	287	12	374.61	24	96%	2	6
c202.50	1024	12	384.14	29	57%	2.41	366
c202.100	3558	11	310.98	30	30%	2.72	18420
c203.25	410	12	390.98	24	96%	2	19
c203.50	1669	11	379.63	29	57%	2.63	1820
c203.100	5701	11	343.21	30	30%	2.72	84107
c204.25	525	10	382.41	25	100%	2.5	33
c204.50	2129	11	356.24	30	60%	2.72	4330
c204.100	7319	11	324.95	31	31%	2.81	225525
c205.25	140	10	341.01	24	96%	2.4	1
c205.50	548	10	322.18	28	56%	2.8	45
c205.100	2047	10	290.10	30	30%	3	2137
c206.25	166	11	367.20	25	100%	2.27	1
c206.50	718	11	349.79	29	57%	2.63	99
c206.100	2641	11	289.76	31	31%	2.81	5056
c207.25	282	10	377.55	24	96%	2.4	11
c207.50	1066	12	370.71	30	60%	2.5	549
c207.100	3078	11	313.94	31	31%	2.81	13401
c208.25	210	10	382.41	25	100%	2.5	5
c208.50	882	12	346.22	30	60%	2.5	299
c208.100	3244	11	292.81	31	31%	2.81	15586

Tableau 5.6 – Solutions du problème pour les instances C avec $t_{max} = 220$.

Prob	# Tournées Phase 1	# Tournées Phase 2	Distance parcourue	# Clients desservis	% Clients desservis	# Moyen de clients	Temps d'exécution
c201.25	139	9	293.76	23	92%	2.55	1
c201.50	505	10	320.68	28	56%	2.8	46
c201.100	1804	10	316.57	30	30%	3	1906
c202.25	634	9	303.86	24	96%	2.66	39
c202.50	2461	10	352.97	29	57%	2.9	3707
c202.100	9064	11	326.50	31	31%	2.81	213601
c203.25	1288	9	298.08	24	96%	2.66	251
c203.50	5171	11	386.71	30	60%	2.72	28996
c204.25	1661	9	316.65	25	100%	2.77	562
c204.50	7405	10	333.05	30	60%	3	109046
c205.25	269	9	314.42	25	100%	2.77	3
c205.50	965	10	318.50	29	57%	2.9	195
c205.100	3575	10	286.76	30	30%	3	9275
c206.25	334	9	309.52	25	100%	2.77	8
c206.50	1428	10	358.92	30	60%	3	782
c206.100	5182	11	283.69	31	31%	2.81	34179
c207.25	639	10	340.36	25	100%	2.5	78
c207.50	2518	11	343.31	30	60%	2.72	6107
c207.100	6420	11	297.03	31	31%	2.81	97259
c208.25	416	9	309.52	25	100%	2.77	20
c208.50	1847	10	325.77	30	60%	3	1996
c208.100	6881	11	283.92	31	31%	2.81	110911

Tableau 5.7 – Solutions du problème pour les instances C avec $t_{max} = 240$.

Instances R

Lors de nos expérimentations sur les instances aléatoires R, nous avons pu résoudre une seule instance à 100 clients. Pour le reste des instances, le nombre de tournées générées à la fin de la phase 1 est donné, mais seuls les résultats pour les 25 et 50 premiers clients sont présentés.

Instances RC

Pour les instances RC, les instances RC204 et RC208 génèrent un plus grand nombre de tournées en phase 1 que les autres instances de la même classe. Pour ces instances, nous nous sommes donc limités aux 25 et 50 premiers clients, et seuls le nombre de tournées générées à la fin de la phase 1 est donné.

Instances C

Nous avons rencontré peu de difficultés pour résoudre ces instances (sauf pour C204). Le temps de service d'un client pour cette classe est plus grand, soit 90 plutôt que 10, c'est pourquoi les tests ont été faits avec des t_{max} de 220 et 240.

Ces tests démontrent que si les problèmes sont assez contraints, il est possible de les résoudre de manière optimale dans des temps très raisonnables même s'ils contiennent une centaine de sommets. Ce temps varie entre quelques secondes et deux à trois jours pour les instances les plus difficiles. Nous remarquons que le nombre de clients desservis

et le nombre moyen de clients par tournée croissent au fur et à mesure que l'on augmente la limite de temps t_{max} pour toutes les classes de problèmes. L'algorithme tend donc à regrouper plus de clients dans une tournée afin de faire moins d'allers et retours au dépôt. Toutefois, au delà d'une certaine limite, les solutions se stabilisent, et le nombre de clients desservis au total ne change plus.

5.3 Résultats sur les instances générées à partir de tournées individuelles

Nous avons remarqué que le pourcentage des clients desservis est relativement faible pour les instances de Solomon à 100 clients, mais tend à augmenter pour les instances à 25 et 50 clients. Ceci est dû au fait que l'on a un seul véhicule pour desservir tous les clients. Afin d'avoir une meilleure évaluation du nombre de clients pouvant être desservis par un seul véhicule, nous avons utilisé les meilleures solutions rapportées dans la littérature pour les instances de Solomon. Nous avons généré de nouvelles instances avec les clients contenus dans chaque tournée individuelle au sein de ces solutions (i.e., une instance est associée à une tournée individuelle). Puisqu'une tournée dans une solution avec tournées multiples est constituée de clients dont les fenêtres de temps sont bien distribuées tout au long de l'horizon de planification, il devrait être possible à notre algorithme de desservir une forte proportion de ces clients.

Chaque instance est identifiée par l'instance de Solomon correspondante et le nombre de clients dans la tournée qui nous intéresse (ainsi, RC201.21 signifie une tournée de 21

clients prise dans la meilleure solution connue pour RC201). Ces solutions sont celles rapportées dans Bent et Van Hentenryck (2001), Czech et Czarnas (2002), Rousseau, Gendreau et Pesant (2003).

Nous avons considéré des valeurs différentes de t_{max} pour les deux classes de problèmes R et RC. Ce t_{max} varie de 140 à 90 pour les instances RC et de 100 à 80 pour les instances R, car un plus grand nombre de tournées est généré en phase 1 avec les instances R. Les résultats révèlent que le pourcentage des clients desservis dépasse maintenant les 80 % dans la plupart des cas, et il est même de 100 % dans deux cas (RC205.21 et R205.21).

En général, nous remarquons que plus les problèmes sont contraints, plus l'algorithme est efficace (en terme de temps de calcul). Les jeux de données de Solomon ne possédaient originalement que deux types de contraintes : les fenêtres de temps et la capacité des véhicules. Le fait d'introduire une limite sur le délai de service au dernier client de chaque tournée nous a ainsi permis de contraindre suffisamment ces problèmes pour pouvoir les résoudre par une méthode exacte.

Prob	# Tournées Phase 1	# Tournées Phase 2	Distance parcourue	#Clients desservis	% Clients desservis	# Moyen de clients	Temps d'exécution
$t_{max} = 140$							
rc201.21	1456	4	335.73	20	95%	5	80
rc201.27	2690	5	375.74	24	88%	4.8	743
rc201.26	1114	5	367.61	23	88%	4.59	101
rc201.26	4087	5	321.57	23	88%	4.59	1301
$t_{max} = 110$							
rc202.38	8970	7	441.55	28	73%	4	26364
rc202.30	2546	7	427.36	26	86%	3.71	786
rc202.32	5922	7	389.02	26	81%	3.71	6950
$t_{max} = 110$							
rc203.27	6013	5	313.74	23	85%	4.59	4470
rc203.29	4618	7	452.95	25	86%	3.57	5689
rc203.44	10974	7	379.78	30	68%	4.28	113602
$t_{max} = 120$							
rc205.29	5609	6	358.87	25	86%	4.16	7588
rc205.22	1338	5	370.62	20	90%	4	86
rc205.28	3687	6	443.27	24	85%	4	938
rc205.21	4702	5	307.27	21	100%	4.2	1450
rc206.32	17700	6	453.37	28	87%	4.66	71160
rc206.35	11487	6	426.42	27	77%	4.5	88764
rc206.33	11908	6	348.61	27	81%	4.5	49263
$t_{max} = 110$							
rc207.27	8145	6	371.03	26	96%	4.33	30034
$t_{max} = 100$							
rc207.35	11492	6	405.46	28	80%	4.66	86864
rc207.38	5759	6	416.29	26	68%	4.33	13630
$t_{max} = 90$							
rc208.34	6816	7	407.08	25	73%	3.57	33726
rc208.33	5804	6	360.70	29	87%	4.83	27404
rc208.33	6350	6	436.73	26	78%	4.33	25298
rc204.34	8768	7	386.61	28	82%	4	36034
rc204.32	11820	7	412	29	90%	4.14	170732
rc204.34	6062	6	413.97	27	79%	4.5	25252

Tableau 5.8 – Solutions pour les instances RC

Prob	# Tournées Phase 1	# Tournées Phase 2	Distance parcourue	# Clients desservis	% Clients desservis	# Moyen de clients	Temps d'exécution
$t_{max} = 100$							
r202.33	3134	7	335.60	25	75%	3.57	1030
r202.32	5375	7	399.33	30	93%	4.28	6833
r205.21	774	6	377.73	21	100%	3.5	80
$t_{max} = 90$							
r205.39	2518	9	401.14	31	79%	3.44	2524
r205.40	4236	7	361.59	30	75%	4.28	15021
r209.31	4987	6	336.16	26	83%	4.33	14683
r209.31	7705	8	454.17	28	90%	3.5	25069
r209.38	10090	9	416	34	89%	3.77	61345
r202.35	13785	8	326.87	29	82%	3.62	87482
$t_{max} = 80$							
r211.48	13646	8	396.2	33	68%	4.12	301214
r211.52	9443	11	392	36	69%	3.27	84848
$t_{max} = 70$							
r204.48	8394	10	388.86	36	75%	3.6	114498
r204.52	5986	10	418.71	32	61%	3.2	45085
r207.55	10145	9	398.48	37	67%	4.11	203275
r207.45	6368	8	416.10	31	68%	3.87	33598

Tableau 5.9 – Solutions pour les instances R

CONCLUSION

Avant d'évoquer les perspectives de recherche ouvertes par ce travail, nous commençons par faire une synthèse de nos contributions.

Nous nous sommes intéressés au problème de livraison à domicile de denrées périssables, formulé comme un problème de tournées de véhicules avec réutilisation de véhicules, dans le cas où un seul véhicule est disponible. Deux objectifs concurrents ont été considérés, soit la minimisation des coûts de transport et la maximisation du profit obtenu en visitant des clients. Pour cela, nous avons adapté un algorithme exact pour le problème du plus court chemin élémentaire avec contraintes de ressources. Cet algorithme nous a permis de résoudre des problèmes avec une centaine de clients.

Plusieurs extensions de notre travail sont possibles. Dans un premier temps, il serait intéressant de résoudre le problème plus général où plusieurs véhicules sont disponibles en résolvant le problème de mise en boîte avec contraintes d'incompatibilité entre les tournées. Ensuite, il serait intéressant de s'attaquer à une version dynamique du problème, qui est plus proche de la réalité, et où les requêtes des clients arrivent en temps réel. Dans un tel contexte, il faut affecter un intervalle de temps pour le service de façon dynamique à chaque fois qu'une nouvelle requête arrive. Il faut aussi mettre à jour les routes planifiées pour répondre aux demandes des clients dans les délais requis. La demande plus grande pour certaines fenêtres de temps, l'incertitude dans les demandes et dans les temps de parcours, et la présence de fenêtres de temps serrées sont des facteurs qui compliquent la tâche. Lorsqu'une nouvelle requête arrive, un traitement préalable est donc requis pour

déterminer si on peut l'accepter ou non. Une stratégie d'acceptation ou de refus pourrait être basée sur une tarification dynamique des fenêtres de temps disponibles. Ceci pourrait permettre aux épiceries en ligne de décroître leurs coûts des opérations en ne sélectionnant que les fenêtres de temps les moins coûteuses et en facturant davantage les clients qui sont prêts à payer plus cher afin d'être livrés dans l'intervalle de temps qui leur convient. C'est dans cet axe que s'orientent les recherches actuelles (Campbell et Savelsbergh 2003).

BIBLIOGRAPHIE

ASDEMIR, K., VARGHESE, S. J. et RAMAYYA, K. (2002) Dynamic Pricing of Home Delivery. *Soumis à Operations Research*.

BENT, R. et VAN HENTENRYCK, P. (2001) A Two-Stage Hybrid Local Search for The Vehicle Routing Problem with Time Windows. *Rapport Technique, CS-01-06, Département d'Informatique, Brown University*.

BERTSIMAS, D. et SIMCHI-LEVI, D. (1994) The New Generation of Vehicle Routing Research : Robust Algorithms Addressing Uncertainty. *Operations Research* 44, 286–304.

BRÄYSY, O. et GENDREAU, M. (2001) Metaheuristics for the Vehicle Routing Problem with Time Windows. *SINTEF Applied Mathematics, Norway*.

CAMPBELL, A. M. et SAVELSBERGH, M. (2002) Decision Support for Consumer Direct Grocery Initiatives. *Rapport TLI-02-09, Georgia Institute of Technology, U.S.A.*

CAMPBELL, A. M. et SAVELSBERGH, M. (2003) Incentive Schemes for Attended Home Deliveries. *Soumis à Transportation Science*.

CZECH, Z. J. et CZARNAS, P. (2002) A Parallel Simulated Annealing for the Vehicle Routing Problem with Time Windows. *Proc. 10 th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, 376–383*.

DROR, M. (1994) Note on the complexity of the shortest path models for columns gene-

ration in VRPTW. *Operations Research* 42, 977–978.

ECOMLOG (2001) <http://www.tuta.hut.fi/ecomlog/>.

FEILLET, D.(2001) Problèmes de tournées avec gains : étude et application au transport inter-usines. *Thèse de Doctorat, Ecole Centrale de Paris*.

FEILLET, D., DEJAX, P. GENDREAU, M. et GUEGUEN, C. (2000) An exact Algorithm for the elementary Shortest path problem with resource constraints : Application to some Vehicle Routing Problems. *Rapport Technique CRT-2000-15, Centre de recherche sur les transports, Montréal*.

FEILLET, D., DEJAX, P. et GENDREAU, M. (2001) Traveling Salesman Problems with Profits : An Overview. *Rapport Technique CRT-2001-24, Centre de recherche sur les transports, Montréal*.

FLEISCHMANN, B. (1990) The vehicle routing problem with multiple use of vehicles. *Rapport de travail, Fachbereich Wirtschaftswissenschaften, Universität Hamburg*.

GONDRAN, B.L. et MINOUX, M. (1985) Graphes et algorithmes. *Eyrolles Edition*.

GROCERYWORKS (2003) Site Web GroceryWorks. <http://www.groceryworks.com>.

GUEGUEN, C. (1999) Méthodes de résolution exacte pour les problèmes de tournées de véhicules. *Thèse de Doctorat, Ecole Centrale de Paris*.

HAYS, T., KESKINOCAK, P. et MALCOME DE LÓPEZ, V. (2004) Strategies and Challenges of Internet Grocery Retailing Logistics. *A paraître dans Applications of Supply Chain Management and E-commerce Research in Industry*.

HOLMSTRÖM, J., PUNAKIVI, M., et YRJÖLÄ, H. (2001) Solving the Last Mile Issue : Reception box or Delivery box ? *International Journal of Physical Distribution and Logistics Management* 31, 427–439.

KÄMÄRÄINEN, V. (2003) The impact of investments on e-grocery logistics operations. *Thèse de Doctorat, Helsinki University of Technology, Industrial Engineering and Management, Helsinki, Finland.*

KÄMÄRÄINEN, V. SARANEN, J. et HOLMSTRÖM, J. (2001) The reception box impact on home delivery efficiency in the e-grocery business. *International Journal of Physical Distribution and Logistics Management* 31, 414–426.

KANE, M. (2002) Tesco aims where Webvan flamed.
<http://news.com/2100-1017-3-813011.html>

LARSEN, J. (1999) Parallelization of the Vehicle Routing Problem with Time Windows. *Thèse de Doctorat, Technical University of Denmark, DTU.*

LIN, I. L. et MAHMASSANI, H. S. (2002) Can online grocers deliver ? Some logistics considerations. *Transportation Research Record* 1817, 17–24.

MILLENIUM (2003) Tesco.com : A millenium Group case study . www.millenn.co.uk.

PARTYKA, J. G. et HALL, R. W. (2000) On the road to service. *OR/MS Today*, Août, 26-30.

PEAPOD. (2002) *Peapod Inc* <http://www.peapod.com/corpinfo/peapodFacts.pdf>.

PROSSER, P., KILBY, P. et SHAW, P. (1998) Dynamic VRPs : A study of scenarios. *Rapport technique, Industrial Engineering and Management Departement, Helsinki, Finland.*

PUNAKIVI, M. (2001) The Value Added of Route Optimization in Home Delivery. *Rapport Technique, Industrial Engineering and Management Departement, Helsinki, Finland.*

PUNAKIVI, M. et SARANEN, J. (2001) Identifying the success factors in e-grocery home delivery. *International Journal of Retail and Distribution Management* 29, 156–163.

ROCHAT, Y. et TAILLARD, E. (1995) Probabilistic Diversification and Intensification in Local Search for Vehicule Routing. *Journal of Heuristics* 1, 147–167.

ROUSSEAU, L.M., GENDREAU, M. et PESANT, G. (2003) Using Constraint-Based Operators to Solve the Vehicle Routing Problem with Time Windows. *Rapport technique CRT-2003-10, Centre de recherche sur les transports, Montréal.*

SARANEN, J. et SMÁROS, J. (2001) An Analytical Model for Home Delivery in the New Economy. *Rapport interne, ECOMLOG.*

SODHI, M. S. (1999) Supply Chain Approaches for Online Grocery Stores. *OR/MS Today*, Juin.

SOLOMON, M. (1987) Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints. *Operations Research* 35, 254–265.

TAILLARD, E.D., LAPORTE, G. et GENDREAU, M. (1996) Vehicule routing with multiple use of vehicules. *Journal of the Operational Research Society* 47, 1065-1070.

YRJÖLÄ, H. (2001) Physical distribution considerations for electronic grocery shopping. *International Journal of Physical Distribution and Logistics Management* 31, 746–761.

YRJÖLÄ, H., TANSKANEN, K. et HOLMSTRÖM, J. (2002) The Way to Profitable Internet Grocery Retailing - six Lessons Learned. *A paraître dans International Journal of Retail and Distribution Management.*

