



HAL
open science

Modèles et méthodes d'optimisation combinatoire pour la conception de chaînes logistiques et l'optimisation des transports

Olivier Péton

► **To cite this version:**

Olivier Péton. Modèles et méthodes d'optimisation combinatoire pour la conception de chaînes logistiques et l'optimisation des transports. Recherche opérationnelle [math.OC]. Université de Nantes - Faculté des Sciences et Techniques, 2014. tel-01157798

HAL Id: tel-01157798

<https://hal.science/tel-01157798v1>

Submitted on 28 May 2015

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

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

UNIVERSITÉ DE NANTES

ÉCOLE DOCTORALE SCIENCES ET TECHNOLOGIES DE
L'INFORMATION ET MATHÉMATIQUES

**Modèles et méthodes d'optimisation
combinatoire pour la conception de chaînes
logistiques et l'optimisation des transports**

Mémoire présenté le 20 novembre 2014 par

Olivier PÉTON

en vue de l'obtention de l'Habilitation à Diriger des Recherches

devant le jury composé de

Van-Dat CUNG	Professeur, Grenoble-INP	Rapporteur
Frédéric SEMET	Professeur, École Centrale de Lille	Rapporteur
Roberto WOLFLER CALVO	Professeur, Université Paris 13 Nord	Rapporteur
Giuseppe BRUNO	Professeur, Università di Napoli Federico II	Examineur
Xavier GANDIBLEUX	Professeur, Université de Nantes	Examineur, Président du Jury
Caroline PRODHON	MdC, Université de Technologie de Troyes	Examinatrice

Habilitation préparée au sein de l'Institut de Recherche en Communications et
Cybernétique de Nantes (UMR CNRS 6597)

Table des matières

1	Introduction générale	9
1.1	Contexte académique	9
1.2	Thématiques de recherche	9
1.3	Organisation du manuscrit	10
2	Heuristiques performantes pour un problème riche de localisation	11
2.1	Introduction aux problèmes de conception des réseaux logistiques	11
2.2	Un modèle riche pour la conception de chaînes logistiques	12
2.3	Heuristique H1 basée sur la relaxation linéaire	17
2.4	Heuristique H2 basée sur la programmation DC	18
2.4.1	Programmation DC et algorithme DCA pour l'optimisation continue	18
2.4.2	Application de DCA au problème (\mathcal{P})	20
2.5	Heuristique H3 basée sur la relaxation lagrangienne	22
2.5.1	Décomposition du problème	22
2.5.2	Schéma général de l'heuristique H3	26
2.6	Evaluation des trois heuristiques	27
2.7	Conclusions et pistes de recherche	28
3	Transports à la demande	31
3.1	Introduction	31
3.2	Projet Marika : une recherche tabou pour la résolution d'un open-VRPTW	32
3.2.1	Description du problème et modélisation	32
3.2.2	Positionnement dans la littérature	33
3.2.3	Recherche tabou pour le problème de transport de personnes handicapées	33
3.3	Projet cTouVeR : conception de Tournées de Véhicules Régulières	34
3.3.1	Modélisation de la régularité de service	34
3.3.2	Le Time-Consistent VRP	35
3.3.3	Une recherche à voisinage large pour le TCVRP	37

3.3.4	Un sous-problème inédit : le VRPTW à fenêtres multiples et sans temps d'attente (VRPmTW-nw)	38
3.3.5	Tests numériques	39
3.4	Projet MCDARP : Problème de type Dial-A-Ride multi-critères	40
3.4.1	Description du problème mutualisé de transport	40
3.4.2	Un modèle mathématique pour le MCDARP	44
3.4.3	Méthode de résolution	46
3.4.4	Expérimentations numériques	47
3.5	Conclusions	48
4	Problèmes de transport avec transferts de passagers	49
4.1	Introduction	49
4.2	Le problème de collectes et livraisons avec navettes (PDPS)	50
4.2.1	Modélisations basées sur les chemins	51
4.2.2	Branch-and-Cut-and-Price	53
4.2.3	Résultats numériques	53
4.3	Le problème de collectes et livraisons avec transferts (PDPT)	54
4.3.1	Une recherche adaptative à voisinage large (ALNS) pour le PDPT	55
4.3.2	Réalisabilité d'une insertion de requête dans une solution	56
4.3.3	Résultats numériques	56
4.4	Dial a Ride Problem avec Transfert (DARPT)	57
4.5	Conclusions	59
5	Projets de recherche actuels	61
5.1	Recherche à voisinage large pour un problème de localisation	61
5.2	État de l'art : problèmes de localisation et développement durable	62
5.3	Projet ORLoGES	63
5.4	Transport urbain mixte passagers/marchandises	64
5.5	Temix β	67
5.6	Projet Végésupply	68
5.7	Thèse 4S Network	69
6	Conclusion générale	71
6.1	Choisir de ne pas choisir...	71
6.2	Perspectives futures	71
6.3	Un métier en évolution	72

7 Curriculum Vitae	75
7.1 Etat Civil et coordonnées	75
7.2 Formation et parcours professionnel	75
7.3 Activités d'encadrement	76
7.3.1 Encadrements de doctorat	76
7.3.2 Autres encadrements	76
7.3.3 Jury de thèses de doctorat	77
7.3.4 Comités de suivi de thèse	77
7.3.5 Jury de thèses de master	77
7.4 Implication dans la communauté scientifique	77
7.4.1 Organisation de congrès scientifiques	77
7.4.2 Rapporteur pour des revues internationales	78
7.4.3 Rapporteur de projets	78
7.4.4 Animation de la recherche	78
7.4.5 Affiliation à des sociétés savantes et groupes de travail	78
7.4.6 Séjours académiques	79
7.5 Subventions de recherche et contrats	79
7.6 Participation à des projets de recherche	79
7.7 Activités d'enseignement	81
7.7.1 Cours enseignés	81
7.7.2 Responsabilités pédagogiques	84
7.7.3 Autres activités liées à l'enseignement	84
7.8 Autres activités	85
A Bibliographie personnelle	87
A.1 Articles publiés dans des revues avec comité de lecture	87
A.2 Articles soumis dans des revues avec comité de lecture	87
A.3 Articles publiés dans des actes de congrès avec comité de lecture	88
A.4 Congrès internationaux avec comité de sélection	88
A.5 Autre conférences internationales	89
A.6 Conférences nationales sans actes	89
A.7 Rapports de recherche	90
A.8 Autres publications nationales	91
A.9 Conférences invitées	91
A Sélection d'articles	101

Résumé

Les travaux décrits dans ce mémoire d'habilitation à diriger des recherches s'articulent autour de deux thématiques principales : les problèmes de localisation de sites dans les chaînes logistiques et les problèmes d'optimisation de tournées de véhicules pour des activités de transport à la demande.

Les problèmes de localisation sont un élément clé de la construction des réseaux logistiques. Cette discipline se situe à l'intersection du management, de la stratégie industrielle, de la logistique et des méthodes quantitatives telles que la recherche opérationnelle. Bien que les problèmes de localisation soient NP-difficiles, il existe aujourd'hui des méthodes efficaces pour résoudre optimalement certains de ces problèmes jusqu'à des tailles d'instance réalistes. Lorsque la taille des instances grandit, il est généralement possible d'obtenir de bonnes bornes inférieures ou supérieures de la solution optimale. Les problèmes de localisation fournissent donc un terrain de jeu idéal pour développer des méthodes de résolution et tester leur passage à l'échelle.

Nous mesurons l'efficacité d'une méthode de résolution selon deux critères : l'écart par rapport à la solution optimale et le temps de calcul. L'enjeu est ici de trouver des solutions optimales ou quasi-optimales en un temps de calcul compatible avec les attentes des utilisateurs. Nous proposons un modèle riche de conception de réseau logistique et plusieurs méthodes heuristiques de résolution. Le modèle en question comporte de nombreuses caractéristiques qui le rendent réaliste : décision échelonnée sur plusieurs périodes, ouvertures et fermetures progressives de sites, prise en compte de plusieurs produits et de leurs liens de nomenclature, prise en compte des fournisseurs et de la sous-traitance.

Nous étudions ensuite extensions des problèmes de tournées de véhicules, difficiles à la fois d'un point de vue théorique et pratique. Tout d'abord, nous construisons des tournées multi-périodiques offrant des horaires de service réguliers aux utilisateurs, alors même que la demande n'est pas régulière. Ensuite, nous résolvons un problème de collecte et livraison en agrégeant cinq critères d'optimisation. Enfin, nous étendons le problème de collectes et livraisons et le dial-a-ride problem en autorisant le transfert de passagers d'un véhicule à un autre. Les applications visées par ces études concernent le transport de personnes handicapées ou le transport scolaire.

Pour ces problèmes, les solveurs n'arrivent pas à résoudre optimalement des instances même de petite taille. L'enjeu est donc double : concevoir des méthodes de résolution exacte pouvant résoudre des instances aussi grandes que possible, et concevoir des heuristiques résolvant des instances réelles en un temps acceptable. Nous décrivons à cet effet plusieurs méthodes de résolution, reposant sur des techniques de branch-and-cut-and-price ou de recherche à voisinage large.

Nous terminons notre exposé en évoquant des projets de recherche actuels mêlant les thématiques de localisation et d'optimisation des transports.

MOTS CLÉS : Optimisation combinatoire, localisation, problèmes de tournées de véhicules,

Introduction générale

1.1 Contexte académique

Ce mémoire d'habilitation à diriger des recherches synthétise mes travaux de recherche menés au sein de l'équipe *Systèmes Logistiques et de Production* (SLP) de l'*Institut de Recherche en Communications et Cybernétique de Nantes* (IRCCyN, UMR CNRS 6597). L'équipe SLP a été construite autour d'une thématique métier : l'optimisation des opérations de planification, production, maintenance, logistique et transport. Ses chercheurs ne sont donc pas focalisés sur une approche de résolution en particulier, même si les techniques de recherche opérationnelle prennent une place prépondérante dans les activités de l'équipe. Le plus petit dénominateur commun dans les travaux évoqués dans ce manuscrit est l'utilisation de modèles de programmation linéaire à variables mixtes. Selon les cas, ces modèles sont résolus à l'aide de méthodes d'optimisation combinatoire exactes, heuristiques, voire matheuristiques.

Pour des raisons de cohérence thématique, seuls des travaux menés depuis 2005 sont décrits dans ce manuscrit. Depuis cette date, je travaille au *Département Automatique-productive* de l'*Ecole des Mines de Nantes* (EMN), en tant que maître assistant. Les liens historiques entre les Ecoles des Mines et l'industrie transparaissent au travers de mes trois missions principales : enseignement, recherche et valorisation de la recherche. Mes activités d'enseignement sont résumées dans le CV détaillé figurant au chapitre 7 de ce manuscrit, de même que les activités de valorisation non évoquées dans les chapitres précédents.

1.2 Thématiques de recherche

Mes travaux s'articulent autour de deux thématiques principales : les problèmes de localisation de sites dans les chaînes logistiques et les problèmes d'optimisation de tournées de véhicules pour des activités de transport, et plus particulièrement de transport à la demande.

Les problèmes de localisation sont un élément clé de la construction des réseaux logistiques. Cette discipline se situe à l'intersection du management, de la stratégie industrielle, de la logistique et des méthodes quantitatives telles que la recherche opérationnelle. La maîtrise des réseaux logistiques est indispensable à toute entreprise industrielle visant un haut niveau de compétitivité sur ses coûts, ses délais ou la qualité de service à ses clients. Cette maîtrise est rendue encore plus complexe dans un environnement économique caractérisé par la mondialisation des échanges et l'accélération des cycles industriels. La presse spécialisée regorge d'exemples de réorganisations, reconfigurations, fusions et mutualisations de réseaux logistiques. Ces mutations industrielles concernent aujourd'hui même les activités industrielles ayant les plus longs cycles de vie, comme les processus continus.

Ceux-ci sont en effet impactés non seulement par les réalités économiques, mais aussi par l'influence de nouvelles législations relatives aux risques industriels ou à l'environnement.

Bien que les problèmes de localisation soient NP-difficiles, il existe aujourd'hui des méthodes efficaces pour résoudre optimalement certains de ces problèmes jusqu'à des tailles d'instance réalistes. Lorsque la taille des instances grandit, il est généralement possible d'obtenir de bonnes bornes inférieures ou supérieures de la solution optimale. Les problèmes de localisation fournissent donc un terrain de jeu idéal pour développer des méthodes de résolution et tester leur passage à l'échelle. Nous mesurons l'efficacité d'une méthode de résolution selon deux critères : l'écart par rapport à la solution optimale et le temps de calcul. L'enjeu est ici de trouver des solutions optimales ou quasi-optimales en un temps de calcul compatible avec les attentes des utilisateurs.

Les problèmes de localisation se situent au niveau stratégique de décision. La portée temporelle des décisions prises est généralement de plusieurs années. Il est donc généralement admis que le temps de calcul des méthodes de résolution puisse être très élevé. Il existe pourtant plusieurs raisons de vouloir déroger à cette règle. Tout d'abord, les décideurs peuvent vouloir tester plusieurs scénarios en un temps assez restreint (par exemple le temps d'une séance de travail). Il est donc appréciable de pouvoir exécuter un scénario en quelques minutes. D'autre part, dans certaines applications, les décisions de localisation doivent être prises assez rapidement. On peut citer comme exemple le positionnement dynamique de flottes d'ambulances, la logistique militaire ou humanitaire. En règle générale, plus les décisions de localisation sont rapprochées dans le temps, plus les décideurs souhaitent obtenir rapidement de bonnes solutions.

Les problèmes d'optimisation des tournées de véhicules évoqués dans ce manuscrit sont difficiles à la fois d'un point de vue théorique et pratique. Ce mémoire aborde notamment plusieurs problèmes nouveaux, tels que les problèmes de transport à la demande avec transferts, pour lesquels la littérature scientifique est encore relativement pauvre. Pour ces problèmes, les solveurs n'arrivent pas à résoudre optimalement des instances même de petite taille. L'enjeu est donc double : concevoir des méthodes de résolution exacte pouvant résoudre des instances de taille réaliste, et concevoir des heuristiques résolvant des instances de plus grande taille, dans un temps acceptable.

1.3 Organisation du manuscrit

Le chapitre 2 introduit un modèle riche de conception de réseau logistique et présente plusieurs méthodes heuristiques de résolution. Le modèle en question ne correspond pas à une entreprise précise, mais comporte de nombreuses caractéristiques qui le rendent réaliste : décision échelonnée sur plusieurs périodes, ouvertures et fermetures progressives de sites, prise en compte de plusieurs produits et de leurs liens de nomenclature, prise en compte des fournisseurs et de la sous-traitance, etc. Ainsi, mettre au point des algorithmes performants pour ce type de problème contribue à rendre possible leur intégration dans les outils logiciels en développement.

Les travaux de recherche en optimisation des transports décrits dans les chapitres 3 et 4 ont émergé à l'issue d'un contrat de valorisation sur la thématique du transport de personnes handicapées. Ce contrat n'avait donné lieu à aucune publication en revue, mais a permis d'identifier plusieurs aspects métier non pris en compte dans les logiciels existants et ignorés des travaux de recherche de l'époque. Cela illustre comment la recherche appliquée peut se nourrir de problématiques réelles.

Le chapitre 5 décrit des projets non encore finalisés. Il est à noter que plusieurs de ces projets comportent à la fois un volet recherche lié au développement de méthodes de résolution efficaces et un volet valorisation lié à la réalisation d'un outil logiciel. Ce chapitre montre comment plusieurs de ces projets ont permis de connecter des travaux en localisation et en transport.

Enfin, le chapitre 6 tire des conclusions et expose quelques perspectives ouvertes par les travaux présentés. La fin de ce mémoire est également l'occasion d'exprimer quelques réflexions personnelles sur mon parcours professionnel, l'évolution des métiers de la recherche, mais aussi les multiples opportunités qui s'offrent aujourd'hui pour la recherche opérationnelle.

Heuristiques performantes pour un problème riche de localisation

2.1 Introduction aux problèmes de conception des réseaux logistiques

L'étude des problèmes de localisation trouve sa source dès la première moitié du XVII^{ème} siècle, avec les travaux de Cavalieri, Fermat ou Torricelli. Leur application aux problèmes industriels a été mentionnée pour la première fois par Weber et Peik (1909). Il était alors question de localiser une seule usine. Il a fallu attendre encore une cinquantaine d'années pour voir le véritable essor des problèmes de localisation, avec la formulation discrète du problème p -médiann (Hakimi, 1964) et les travaux contemporains de Manne, Balinsky, Kuehn et Hamburger.

Depuis une quinzaine d'années, on voit fleurir des modèles de plus en plus riches, c'est-à-dire intégrant un grand nombre de caractéristiques réalistes :

- la présence de plusieurs échelons dans la chaîne logistique, par exemple fournisseurs, usines, entrepôts, clients,
- la prise en compte de plusieurs produits, éventuellement liés par des liens de nomenclature,
- des flux complexes de produits entre les différentes unités, notamment des livraisons directes de certains clients depuis les usines, des flux inverses,
- le choix entre plusieurs niveaux de technologie, plusieurs modes de transport,
- la prise en compte de décisions tactiques, notamment de stockage,
- une modélisation des coûts de plus en plus réaliste,
- la prise en compte de facteurs environnementaux.

On appelle Supply Chain Network Design (SCND) l'ensemble des modèles riches généralisant le problème de localisation à l'ensemble d'une chaîne logistique. Il existe un grand nombre de variantes de SCND. Selon ces variantes, les variables de décision peuvent concerner le nombre, la localisation et la capacité de sites, les flux de matières produites, transformées, stockées et transportées dans le réseau. Dans les applications logistiques, la quasi-totalité des modèles concernent des localisations discrètes. Il contiennent donc des variables binaires de localisation, c'est-à-dire valant 1 lorsqu'un site candidat est sélectionné et 0 sinon.

Les problèmes de localisation et de SCND ont donné lieu à plusieurs états de l'art récents : Beamon (1998), Owen et Daskin (1998), Daskin *et al.* (2005), Meixell et Gargeya (2005), Sahin et Süral (2007), Melo *et al.* (2009). On citera également les états de l'art de Akçali *et al.* (2009), Aras *et al.* (2010) et Pati *et al.* (2013) pour les modèles de logistique inverse ou décrivant des réseaux logistiques en boucle fermée.

Lorsqu'il n'y a qu'un site à localiser parmi n sites candidats, le problème revient à une comparaison de n scénarios indépendants. Lorsqu'il y a plusieurs sites à localiser, et a fortiori des contraintes ou caractéristiques additionnelles venant enrichir le modèle, le problème devient difficile. Contrairement aux problèmes d'optimisation des transports qui seront abordés dans les chapitres suivants, les meilleurs solveurs de PLNE peuvent aujourd'hui résoudre efficacement des instances de taille moyenne de SCND. Ces problèmes constituent donc un excellent laboratoire pour évaluer la performance des heuristiques et tester leur passage à l'échelle.

Ce chapitre repose largement sur les travaux de thèse de doctorat de Phuong Nga Thanh (Thanh, 2008) menés entre 2005 et 2008, et que j'ai co-encadrés avec Nathalie Bostel. Nous proposons un modèle riche pour une chaîne logistique avec échelons multiples et produits multiples. Le modèle obtenu est un programme linéaire mixte en nombres entiers pour lesquels les principaux solveurs montrent leurs limites lorsque la taille des instances grandit. Nous avons donc développé trois méthodes heuristiques basées respectivement sur la relaxation linéaire (section 2.3), la programmation DC (section 2.4) et la relaxation lagrangienne (section 2.5).

2.2 Un modèle riche pour la conception de chaînes logistiques

Nous présentons ici le modèle mathématique qui sera résolu par les 3 approches décrites dans les sections suivantes (Thanh *et al.*, 2008). Il s'agit d'un modèle stratégique multi-périodes pour localiser les sites de production et de stockage de produits non périssables, sur un horizon stratégique de plusieurs périodes (saisons, années). Le réseau logistique étudié comporte quatre types d'acteurs : les fournisseurs de rang 1, les sites de production (usines), les sites de distribution (entrepôts) et les clients. Ce réseau assure la production et la distribution d'un ensemble de produits qui peuvent être manufacturés en plusieurs étapes dans différentes usines. La figure 2.1 illustre la structure générale du réseau logistique.

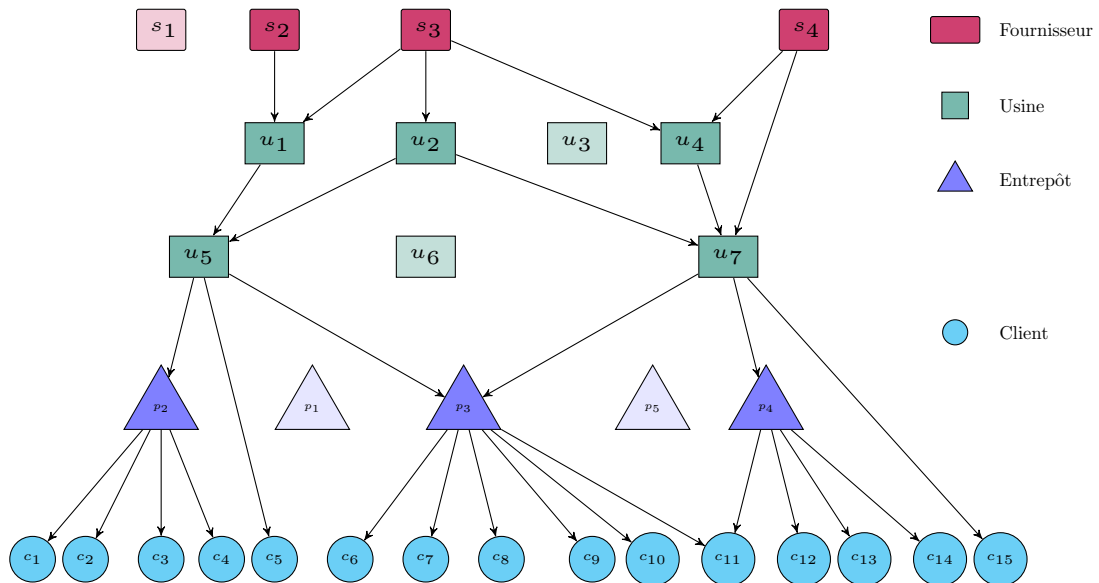


Figure 2.1 – Le réseau logistique étudié

La figure montre plusieurs caractéristiques du modèle : les fournisseurs peuvent livrer plusieurs usines, et chaque usine peut être livrée par plusieurs fournisseurs ; un produit peut nécessiter le passage par plusieurs sites de production ; il peut exister des livraisons directes d'une usine à un client ; les clients peuvent être livrés par plusieurs entrepôts (pas de *single sourcing*).

En plus des décisions d'ouverture ou fermeture de site, le modèle permet l'agrandissement de sites existants. On suppose à cet effet l'existence d'*options de capacité*, c'est-à-dire d'un ensemble discret de capacités supplémentaires qu'on peut ajouter à un site existant. Ces options de capacité sont cumulables : on peut donc modéliser plusieurs agrandissements d'un même site. On distingue deux types d'entrepôts. Les entrepôts dits privés appartiennent à l'entreprise : une fois ouverts, ils ne peuvent être refermés pendant l'horizon de planification ; une fois fermés, ils ne peuvent être ré-ouverts. Les entrepôts publics peuvent être ouverts ou fermés toutes les 2 périodes, durée considérée ici comme la durée minimale d'un contrat de location.

Le modèle repose sur les notations suivantes :

Ensembles

T	horizon de planification
S	fournisseurs potentiels
I	sites potentiels pour les usines
J	sites potentiels pour les entrepôts
$J^* \subset J$	entrepôts appartenant à l'entreprise (ou privés)
C	clients
O	options de capacités
P_r	matières premières (produits livrés par les fournisseurs)
P_f	produits finis
P_m	produits manufacturés semi-finis ou finis ($P_f \subset P_m$)
P	tous les produits, $P = P_r \cup P_m$
$\Gamma^+(i)$	ensemble des successeurs du site i (au sens du graphe de la Figure 2.1)
$\Gamma^-(i)$	ensemble prédécesseurs du site i (au sens du graphe de la Figure 2.1)

On définit enfin deux sous-ensemble d'usines et d'entrepôts : I_c et J_c^* représentent les sites susceptibles de fermer, I_o , J_o^* représentent les sites susceptibles d'ouvrir.

Paramètres

$A_{s,p}^t$	= 1 si le fournisseur $s \in S$ peut fournir la matière première $p \in P_r$ à la période $t \in T$, 0 sinon
$A_{i,o}^t$	= 1 si l'option de capacité o peut être ajoutée au site $i \in I \cup J^*$ à la période t , 0 sinon
$A_{i,p}^t$	= 1 si l'usine $i \in I$ peut produire le produit $p \in P_m$ à la période t , 0 sinon
$A_{j,p}^t$	= 1 if l'entrepôt $j \in J$ peut stocker $p \in P_f$ à la période t , 0 sinon

Capacités

$K_{s,p}^t$	quantité maximale de matière première p disponible chez le fournisseur s la période t
K_i	capacité initiale du site $i \in I \cup J$
\bar{K}_i	capacité maximale installable au site $i \in I \cup J$
K_o	capacité de l'option o
\underline{U}_i	pourcentage minimal d'utilisation du site $i \in I \cup J^*$ (indépendant du temps)
\bar{U}_i^t	pourcentage maximal d'utilisation du site $i \in I \cup J^*$ à la période t
KS_j	stock initial à l'entrepôt j
KS_o	capacité de stockage dans l'option o (si o est ajoutée à un entrepôt)

Clients

$D_{c,p}^t$	demande supposée connue de produit p par le client c à la période t
-------------	---

Produits

$SubC_p$	= 1 si la production de $p \in P_m$ peut être sous-traitée, 0 sinon
$BoM_{p,p'}$	quantité de $p \in P_r \cup P_m$ dans une unité de $p' \in P_m$ (BoM = Bill of Material)
$WP_{p,i}$	charge de travail nécessaire pour traiter une unité de p sur le site $i \in I \cup J$
$WS_{p,j}$	charge de travail liée au stockage d'une unité de p dans l'entrepôt $j \in J$, pendant une période
$\Delta_{i,j}$	nombre de livraisons depuis l'usine i vers l'entrepôt j , pendant une période.

Coûts

CS_s	coût fixe de sélection d'un fournisseur
CO_i	coût fixe d'ouverture du site $i \in I \cup J^*$
CC_i	coût fixe de fermeture de i
$CA_{i,o}$	coût fixe d'ajout de l'option de capacité o sur le site i
CU_i	coût fixe par période pour le site $i \in I \cup J$
$CU_{i,o}$	coût fixe par période si l'option de capacité o est installée sur le site i
$CP_{p,i}$	coût de traitement d'une unité de $p \in P$ par l'usine $i \in I$
$CSub_p$	coût de sous-traitance d'une unité de $p \in P$
$CH_{p,j}$	coût de détention par période d'une unité de p à l'entrepôt $j \in J$
$CW_{p,j}$	coût de traitement d'une unité de p à l'entrepôt $j \in J$
$CT_{p,i,j}$	coût de transport d'une unité de p entre deux sites i et j .

Tous les coûts sont supposés linéaires en fonction des quantités, et sans perte de généralité, indépendants du temps.

Variables binaires

x_i^t	= 1 si le site $i \in S \cup I \cup J$ est actif à la période t , 0 sinon
$y_{i,o}^t$	= 1 si l'option de capacité o est ajoutée au site $i \in I \cup J$ à la période t , 0 sinon
$z_{s,p}^t$	= 1 si le fournisseur s fournit le produit p à la période t , 0 sinon

Variables continues pour les flux de matières

$q_1^t(p, i, j)$	flux de produit p sur l'arc (i, j) à la période t
$q_2^t(p, i)$	quantité de p produits par l'usine i à la période t
$q_3^t(p, i)$	quantité de p sous-traitée par l'usine i à la période t
$q_4^t(p, j)$	quantité de p dans l'entrepôt j au début de la période t

Contraintes

Les contraintes suivantes sont définies pour tout $t \in T$.

$$\sum_{i \in \Gamma^-(c)} q_1^t(p, i, c) \geq D_{c,p}^t \quad c \in C, p \in P_f \quad (2.1)$$

$$q_4^{t-1}(p, j) + \sum_{i \in \Gamma^-(j)} q_1^t(p, i, j) = q_4^t(p, j) + \sum_{c \in \Gamma^+(j)} q_1^t(p, j, c) \quad j \in J, p \in P_f \quad (2.2)$$

$$\sum_{s \in \Gamma^-(i)} q_1^t(p, s, i) \geq \sum_{p' \in P_m} BoM_{p,p'} q_2^t(p', i) \quad i \in I, p \in P_r \quad (2.3)$$

$$\sum_{i' \in \Gamma^-(i)} q_1^t(p, i', i) + q_3^t(p, i) \geq \sum_{p' \in P_m} BoM_{p,p'} q_2^t(p', i) \quad i \in I, p \in P_r \cup P_m \quad (2.4)$$

$$q_2^t(p, i) + q_3^t(p, i) \geq \sum_{j \in \Gamma^+(i)} q_1^t(p, i, j) \quad i \in I, p \in P_m \quad (2.5)$$

Les contraintes (2.1) à (2.5) permettent d'assurer la satisfaction des demandes à chaque échelon du réseau : clients, produits finis aux entrepôts, matières premières livrées par les fournisseurs aux usines, produits semi-finis et produits finis manufacturés ou sous-traités.

$$K_i x_i^t + \sum_{o \in O} K_o y_{i,o}^t \leq \bar{K}_i \quad i \in I \quad (2.6)$$

$$K_j x_j^t + \sum_{o \in O} K_o y_{j,o}^t \leq \bar{K}_j \quad j \in J \quad (2.7)$$

$$\sum_{p \in P_m} WP_{p,i} q_2^t(p, i) \geq \underline{U}_i (K_i x_i^t + \sum_{o \in O} K_o y_{i,o}^t) \quad i \in I \quad (2.8)$$

$$\sum_{p \in P_m} WP_{p,i} q_2^t(p, i) \leq \bar{U}_i (K_i x_i^t + \sum_{o \in O} K_o y_{i,o}^t) \quad i \in I \quad (2.9)$$

$$\sum_{p \in P_f} \sum_{c \in \Gamma^+(j)} WP_{p,j} q_1^t(p, j, c) \geq \underline{U}_j (K_j x_j^t + \sum_{o \in O} K_o y_{j,o}^t) \quad j \in J \quad (2.10)$$

$$\sum_{p \in P_f} \sum_{c \in \Gamma^+(j)} WP_{p,j} q_1^t(p, j, c) \leq \bar{U}_j (K_j x_j^t + \sum_{o \in O} K_o y_{j,o}^t) \quad j \in J \quad (2.11)$$

$$\sum_{p \in P_f} WS_{p,j} \left(q_4^t(p, j) + \sum_{i \in \Gamma^-(j)} \frac{q_1^t(p, i, j)}{\Delta_{i,j}} \right) \leq \bar{U}_j (KS_j x_j^t + \sum_{o \in O} KS_o y_{j,o}^t) \quad j \in J \quad (2.12)$$

Les contraintes (2.6) et (2.7) modélisent le respect des capacités (capacité initiale + options de capacité) des usines et entrepôts. Les contraintes (2.8) à (2.11) précisent les bornes inférieures et supérieures sur le pourcentage d'utilisation des usines et des entrepôts. Les contraintes (2.10) et (2.11) donnent les mêmes informations pour les entrepôts. Les contraintes (2.12) concernent la capacité de stockage aux entrepôts. Par mesure de sécurité, le niveau de stock $q_4^t(p, j)$ est augmenté d'une fois la valeur du stock cyclique.

$$\sum_{i \in \Gamma^+(s)} q_1^t(p, s, i) \leq z_{s,p}^t K_{s,p}^t \quad s \in S, p \in P_r \quad (2.13)$$

$$z_{s,p}^t \leq x_s^t A_{s,p}^t \quad s \in S, p \in P_r \quad (2.14)$$

$$q_1^t(p, i, i') \leq A_{i,p}^t \bar{K}_i \quad i \in I, i' \in \Gamma^+(i), p \in P_m \quad (2.15)$$

$$q_1^t(p, j, c) \leq A_{j,p}^t \bar{K}_j \quad j \in J, c \in \Gamma^+(j), p \in P_f \quad (2.16)$$

Les contraintes (2.13) précisent qu'un fournisseur ne peut livrer un produit que s'il a été sélectionné pour cela. Les contraintes (2.14) précisent quels produits un fournisseur peut livrer. Les contraintes (2.15) et (2.16) imposent qu'une usine ou un entrepôt ne peuvent livrer que des produits qu'ils traitent.

$$q_3^t(p, i) \leq x_i^t \text{SubC}_p M \quad i \in I, p \in P_m \quad (2.17)$$

$$y_{i,o}^t \leq x_i^t A_{i,o}^t \quad i \in I, o \in O \quad (2.18)$$

$$x_i^t \geq x_i^{t+1} \quad i \in I_c \quad (2.19)$$

$$x_i^t \leq x_i^{t+1} \quad i \in I_o \quad (2.20)$$

$$y_{i,o}^t \leq y_{i,o}^{t+1} \quad i \in I, o \in O \quad (2.21)$$

$$x_i^t \geq y_{i,o}^{t-1} \quad i \in I, o \in O \quad (2.22)$$

Les contraintes (2.17) restreignent la sous-traitance aux seules usines ouvertes. Les contraintes (2.18) vérifient si les options de capacité sont installées sur des sites compatibles. Les contraintes (2.19) à (2.22) empêchent une usine de changer de statut (ouvert ou fermé) plus d'une fois dans l'horizon de temps T . Les sites fermés ne ré-ouvrent pas, les sites ouverts ou agrandis ne ferment pas.

$$y_{j,o}^t \leq x_j^t A_{j,o}^t \quad j \in J^*, o \in O \quad (2.23)$$

$$x_j^t \geq x_j^{t+1} \quad j \in J_c^* \quad (2.24)$$

$$x_j^t \leq x_j^{t+1} \quad j \in J_o^* \quad (2.25)$$

$$y_{j,o}^t \leq y_{j,o}^{t+1} \quad j \in J^*, o \in O \quad (2.26)$$

$$x_j^t \geq y_{j,o}^{t-1} \quad j \in J^*, o \in O \quad (2.27)$$

$$-1 \leq 2x_j^t - x_j^{t-1} - x_j^{t+1} \leq 1 \quad j \in J \setminus J^* \quad (2.28)$$

Les contraintes (2.23) à (2.27) empêchent les entrepôts de changer plus d'une fois de statut. Une exception concerne les entrepôts externes qui peuvent changer de statut toutes les 2 périodes (2.28).

$$x_i^t \in \{0, 1\} \quad i \in S \cup I \cup J \quad (2.29)$$

$$y_{i,o}^t, y_{j,o}^t \in \{0, 1\} \quad i \in I, j \in J, o \in O \quad (2.30)$$

$$z_{s,p}^t \in \{0, 1\} \quad s \in S, p \in P_r \quad (2.31)$$

$$q_1^t(p, i, i') \geq 0 \quad p \in P_m, i \in S \cup I \cup J, i' \in \Gamma^+(i) \quad (2.32)$$

$$q_2^t(p, i), q_3^t(p, i) \geq 0 \quad p \in P_m, i \in I \quad (2.33)$$

$$q_4^t(p, j) \geq 0 \quad p \in P_f, j \in J \quad (2.34)$$

Finalement, les contraintes (2.29) à (2.34) décrivent la nature des variables et leur non négativité.

Pour des raisons de clarté, ce modèle est légèrement simplifié par rapport à celui présenté dans Thanh *et al.* (2008). Nous omettons quelques caractéristiques supplémentaires visant à rendre le modèle plus réaliste, ainsi que des inégalités valides. De plus, les contraintes (2.3) et (2.4) sont agrégées dans Thanh *et al.* (2008). Nous les désagréons ici en vue de présenter la relaxation Lagrangienne en section 2.5.

Fonction objectif

La fonction objectif à minimiser est la somme de tous les coûts logistiques sur l'horizon de temps considéré.

$$\begin{aligned} \text{Min } z = & \sum_{t \in T} \sum_{s \in S} CS_s x_s^t && \text{(Sélection de fournisseurs)} \\ & + \sum_{t \in T} \sum_{i \in I_o \cup J_o^*} CO_i (x_i^{t+1} - x_i^t) && \text{(Coût fixe d'ouverture de sites)} \\ & + \sum_{t \in T} \sum_{i \in I_c \cup J_c^*} CC_i (x_i^t - x_i^{t+1}) && \text{(Coût de fermeture des sites)} \\ & + \sum_{t \in T} \sum_{i \in I \cup J^*} \sum_{o \in O} CA_{i,o} (y_{i,o}^{t+1} - y_{i,o}^t) && \text{(Coût des options de capacité)} \\ & + \sum_{t \in T} \sum_{i \in I \cup J^*} CU_i x_i^t + \sum_{o \in O} CU_{i,o} y_{i,o}^t && \text{(Coûts opératoires fixes)} \\ & + \sum_{t \in T} \sum_{p \in P_m} \sum_{i \in I} CP_{p,i} q_2^t(p, i) + CS_{ub_p} q_3^t(p, i) && \text{(Coût de production)} \\ & + \sum_{t \in T} \sum_{p \in P_f} \sum_{j \in J} CH_{p,j} (q_4^t(p, j) + \sum_{i \in \Gamma^-(j)} \frac{q_1^t(p, i, j)}{2\Delta_{i,j}}) && \text{(Coût de stockage)} \\ & + \sum_{t \in T} \sum_{p \in P_f} \sum_{j \in J} \sum_{c \in \Gamma^+(j)} CW_{p,j} q_1^t(p, j, c) && \text{(Coût de distribution aux clients)} \\ & + \sum_{t \in T} \sum_{p \in P} \sum_{i \in S \cup I \cup J} \sum_{j \in \Gamma^+(i)} CT_{p,i,j} q_1^t(p, i, j) && \text{(Coût de transport entre sites)} \end{aligned}$$

Les cinq premières composantes sont les coûts fixes. Notons que les coûts d'ouverture, de fermeture de site ou d'ajout des options de capacité ne sont payés qu'au plus une fois au cours de l'horizon de temps. Les autres termes sont des coûts variables proportionnels à l'activité des différents sites. Notamment, les coûts de stockage dépendent du stock moyen, défini comme la moitié des quantités livrées par les usines aux entrepôts (le coût d'un éventuel stock de sécurité n'est pas pris en compte

dans ce modèle de localisation). Par la suite, on appelle (\mathcal{P}) le problème de conception de chaîne logistique consistant à minimiser la fonction objectif sous les contraintes (2.1)–(2.34).

2.3 Heuristique H1 basée sur la relaxation linéaire

La relaxation linéaire consiste à relaxer les contraintes d'intégralité d'un programme en nombres entiers ou à variables mixtes. Utiliser la relaxation linéaire pour résoudre des problèmes d'optimisation combinatoire n'est pas toujours gage de réussite. Néanmoins, pour certains problèmes tels que les problèmes de localisation, cette technique peut s'avérer efficace pour peu que les procédures d'arrondi des variables relaxées soient performantes.

Nous proposons une heuristique itérative, notée H1, qui consiste à résoudre une suite de relaxations linéaires du MILP initial. À chaque itération, on résout la relaxation linéaire du problème courant. Les variables binaires dont la valeur vaut 0 ou 1 dans la solution optimale sont alors figées à 0 ou 1 et considérées comme des constantes par la suite. Après un certain nombre d'itérations, on suppose que le problème courant sera suffisamment petit pour être résolu en un temps acceptable par un solveur. Le logigramme sur la Figure 2.2 représente l'heuristique H1.

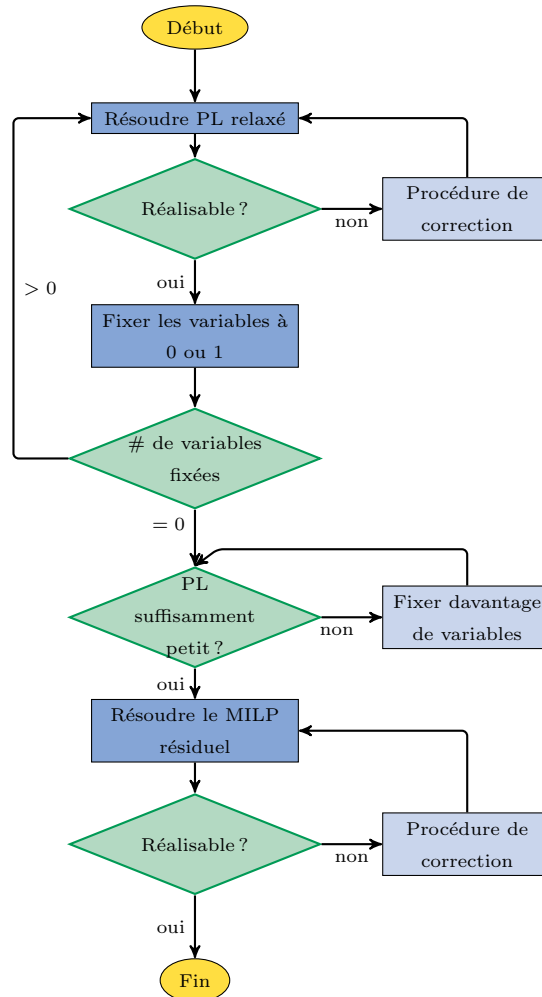


Figure 2.2 – Schéma général de l'heuristique H1

Nous appelons *fixation d'une variable* l'action de lui attribuer la valeur 0 ou 1, et donc de ne plus

la considérer comme une variable par la suite. La *relaxation d'une variable fixée* est l'action de la réintégrer dans le processus et de la considérer de nouveau comme une variable continue entre 0 et 1.

Une mauvaise décision de fixation de variable peut conduire à une non-réalisabilité du problème courant aux itérations suivantes. Dans ce cas, une procédure de correction basée sur le principe de *Feasibility Pump* (Fischetti *et al.* (2005); Bertacco *et al.* (2007)) permet de rétablir la réalisabilité.

Lorsque la relaxation linéaire ne permet plus de nouvelles fixations de variables, on calcule le nombre de variables binaires fixées. L'objectif est de continuer à fixer à 0 ou 1 des variables binaires, jusqu'à ce que le problème résiduel soit suffisamment petit pour être résolu avec un solveur. Si cela n'est pas le cas, il faut fixer davantage de variables binaires. Parmi ces variables binaires, les variables de localisation (variables x) liées aux usines et entrepôts jouent un rôle prépondérant car elles sont associées à des coûts fixes élevés et leur valeur impacte de nombreuses autres variables.

On développe donc des procédures spécifiques permettant de fixer de manière heuristique les variables de localisation à 0 ou à 1. En ce qui concerne les entrepôts, la procédure de fixation fait intervenir deux seuils d'acceptabilité ε_0^* et ε_1^* respectivement proches de 0 et 1. Les variables de localisation inférieures à ε_0^* sont fixées à 0 et celles supérieures à ε_1^* sont fixées à 1. En ce qui concerne les variables de localisation des usines, on ne souhaite pas pallier tous les problèmes de capacité par un coûteux recours à la sous-traitance. L'heuristique de fixation repose sur un principe glouton d'arrondi à 1 des variables fractionnaires les plus élevées. Ces procédures sont décrites dans Thanh *et al.* (2010).

La dernière phase de l'heuristique consiste à résoudre le problème résiduel par une méthode exacte (nous avons utilisé le solveur Xpress-MP). Si la solution obtenue n'est pas réalisable, une procédure de correction est appliquée.

L'heuristique H1 permet de résoudre de manière approchée et en un temps inférieur à 1 heure des instances allant jusqu'à 60 sites candidats et 270 clients. Toutefois, pour des instances de taille supérieure, le temps de calcul croît rapidement. De plus, pour les instances les plus grandes, environ 75% du temps de calcul est consacré à la résolution des relaxations linéaires.

2.4 Heuristique H2 basée sur la programmation DC

Afin de réduire le temps de calcul de l'heuristique H1, nous proposons de remplacer la relaxation linéaire par une méthode plus rapide, à savoir l'algorithme DCA. DCA est un algorithme de recherche locale pour résoudre les problèmes de type DC. Il a été introduit par Pham et El Bernoussi (1988), puis développé par Le Thi et Pham depuis 1994 (Le Thi et Pham, 2005). Avant de présenter l'heuristique, rappelons quelques résultats théoriques liés à la programmation DC (Difference of Convex functions) et à l'algorithme DCA.

2.4.1 Programmation DC et algorithme DCA pour l'optimisation continue

Soit X un ensemble convexe fermé et deux fonctions convexes g et h sur X . Le problème $\text{Min } \{g(x) - h(x) \mid x \in X\}$ est appelé problème D.C (Différence de fonctions Convexes).

Par ailleurs, on rappelle que la fonction conjuguée d'une fonction f dans X s'écrit

$$\begin{aligned} f^*(y) &= \sup_x \{\langle x, y \rangle - f(x) \mid x \in X\}, \\ &= \inf_x \{f(x) - \langle x, y \rangle \mid x \in X\}. \end{aligned}$$

Si f est semi-continue inférieure propre sur X , alors f^* l'est sur l'espace dual de X . De plus, $f^{**} = f$.

Theorem 2.1. *Dualité en programmation DC (Toland, 1978)*

Soient g et h deux fonctions semi-continues inférieures propres. Alors

$$\min_{x \in \text{dom}(g)} \{g(x) - h(x)\} = \min_{y \in \text{dom}(h^*)} \{h^*(y) - g^*(y)\}.$$

De plus, si y^0 est un minimum de $h^* - g^*$ alors tout $x^0 \in \partial g^*(y^0)$ est un minimum de $g - h$ sur X .

Le Théorème 2.1 montre la liaison étroite entre $\{g(x) - h(x) \mid x \in X\}$ et $\{h^*(y) - g^*(y) \mid y \in Y\}$. Ainsi, le problème dual de $\text{Min } \{g(x) - h(x) \mid x \in X\}$ est défini par $\text{Min } \{h^*(y) - g^*(y) \mid y \in Y\}$.

Soient $x^{k+1} \in \text{argmin } \{g(x) - h(x) \mid x \in \partial g^*(y^k)\}$ et $y^k \in \text{argmin } \{h^*(y) - g^*(y) \mid y \in \partial h(x^k)\}$.

L'obtention directe de x^{k+1} et y^k peut être difficile. L'algorithme DCA est une méthode de recherche locale qui construit alternativement deux suites $\{x^k\}$ et $\{y^k\}$ en calculant des sous-gradients des composantes convexes de la fonction objectif du problème primal et ceux du problème dual. Les limites x^* et y^* convergent vers l'optimum local des problèmes primal et dual. L'algorithme 1 décrit le fonctionnement de DCA.

Algorithme 1 *Schéma de la forme simplifiée de DCA*

0. Point initial : $x^0 \in X$.
 1. À chaque itération k , x^k étant connu, **déterminer** $y^k \in \partial h(x^k)$.
 2. **Trouver** $x^{k+1} \in \partial g^*(y^k)$.
 3. Si test d'arrêt vérifié **Stop** ; sinon $k \leftarrow k + 1$ et **aller** en 1.
-

Pour une étude complète de la programmation DC et de l'algorithme DCA, nous renvoyons le lecteur à Le Thi (1997), Pham et El Bernoussi (1988) et Le Thi et Pham (2005).

Application de DCA aux problèmes linéaires mixtes

Considérons le problème linéaire mixte suivant :

$$\begin{aligned} \text{Min } & f(x, y) = c^T x + d^T y \\ & Ax + By \leq C \\ & x \in \{0, 1\}^n \\ & y \in \mathbb{R}^p \end{aligned}$$

La première opération consiste à transformer ce problème en problème DC équivalent, ce qui est possible au vu du théorème de la pénalité exacte :

Theorem 2.2. La pénalité exacte (Le Thi et al., 1999) *Soient C un polyèdre convexe, borné et non vide ; f une fonction concave, finie dans C et p une fonction non négative, concave dans C . Alors il existe $\tau_0 \geq 0$ tel que pour tout $\tau > \tau_0$, (P) et (P_τ) ont le même ensemble de solutions réalisables et la même valeur optimale :*

$$\begin{aligned} \text{Min } \{f(x, y) \mid (x, y) \in C, p(x) \geq 0\} & \quad (P) \\ \text{Min } \{f(x, y) + \tau p(x) \mid (x, y) \in C\} & \quad (P_\tau) \end{aligned}$$

La fonction p est appelée la fonction de pénalité et τ est appelé le coefficient de la fonction de pénalité.

Les contraintes d'intégralité sont remplacées par l'ajout d'une fonction de pénalité dans la fonction objectif. L'ensemble des contraintes devient alors convexe car il ne contient que des contraintes linéaires. En revanche, la fonction objectif n'est plus linéaire. Les fonctions de pénalités usuelles pour des variables binaires sont de type $p(x) = \sum_i \text{Min} \{x_i, 1 - x_i\}$ ou $p(x) = \sum_i x_i(1 - x_i)$. Nous utiliserons la première forme.

Le choix efficace de la décomposition de la fonction objectif dépend de la structure de chaque problème. Néanmoins, nous pouvons remarquer que la construction de y^k se fait directement par le calcul implicite de $\partial h(x^k)$ tandis que la suite x^k est calculée par la résolution du problème $x^{k+1} \in \partial g^*(y^k)$. De ce fait, il est plus avantageux de prendre la fonction g la plus simple possible. Les composantes restantes seront placées dans h .

Soit (C) l'ensemble des contraintes. La décomposition usuelle pour les problèmes MILP consiste à définir $h(x, y) = -c^T x - d^T y - \tau p(x)$ et $g(x, y) = 0$ si $(x, y) \in (C)$, $+\infty$ sinon.

2.4.2 Application de DCA au problème (\mathcal{P})

Modélisation DC du problème (\mathcal{P})

Nous appliquons la théorie de la pénalité exacte pour reformuler le problème (\mathcal{P}) comme un problème DC. Nous relaxons les contraintes d'intégralité (2.29)–(2.31) et ajoutons à la fonction objectif la pénalité définie par

$$\begin{aligned} p(x, y, z) = & \tau \sum_{s \in S} \min\{x_s^t, 1 - x_s^t\} + \tau(i) \sum_{i \in I} \min\{x_i^t, 1 - x_i^t\} + \tau(j) \sum_{j \in J} \min\{x_j^t, 1 - x_j^t\} \\ & + \tau \sum_{i \in I \cup J, o \in O} \min\{y_{i,o}^t, 1 - y_{i,o}^t\} \\ & + \tau \sum_{s \in S, p \in P_r} \min\{z_{s,p}^t, 1 - z_{s,p}^t\} \end{aligned}$$

où $\tau(i)$, $\tau(j)$ et τ sont les coefficients associés aux variables de localisation des usines, des entrepôts et des autres variables binaires.

La formulation DC du modèle (\mathcal{P}) est donc la suivante :

$$\begin{aligned} & \text{Min } z + p \\ & \text{s.c. : (1) - (26),} \\ & x_i^t, y_{i,o}^t, z_{s,p}^t \in [0, 1] \\ & q_1^t(p, i, j), q_2^t(p, i), q_3^t(p, i), q_4^t(p, j) \geq 0 \end{aligned}$$

La fonction objectif est exprimée comme une différence des deux fonctions convexes suivantes : $g(x, y, z, q_1, q_2, q_3, q_4) = 0$ si le vecteur $(x, y, z, q_1, q_2, q_3, q_4)$ est réalisable, et $+\infty$ sinon ; $h(x, y, z, q_1, q_2, q_3, q_4) = -z - p(x, y, z)$.

Les suites $\{X^k\}$ et $\{Y^k\}$ sont ensuite calculées, avec

$$X^k = (x_i^t, y_{i,o}^t, z_{s,p}^t, q_1^t(p, i, j), q_2^t(p, i), q_3^t(p, i), q_4^t(p, j))$$

et

$$Y^k = \partial h(X^k) = -\partial(z) - \partial(p).$$

Le calcul de $\partial h(X^k)$ est détaillé dans Thanh *et al.* (2012).

Schéma général de l'heuristique

Nous notons H2 l'heuristique qui améliore l'heuristique H1 en remplaçant les résolutions de PL relaxés par la résolution de problèmes DC à l'aide de l'algorithme DCA. La Figure 2.3 représente un logigramme de l'heuristique H2.

À la première itération, le point initial fourni à DCA est la solution de la relaxation linéaire du problème initial. C'est la seule résolution de programme linéaire nécessaire à cette heuristique. Aux itérations suivantes, on résout une suite de problèmes DC en repartant de la solution obtenue à l'itération précédente, puis on procède aux fixations de variables binaires. On procède ainsi tant que le problème résiduel est jugé trop grand pour être résolu directement avec un solveur. Une procédure d'accélération est appelée lorsque le nombre de variables fixées à 0 ou 1 à l'itération courante est inférieure à un paramètre ν . Cette procédure consiste à forcer à 1 la variable de localisation d'usine ayant la valeur fractionnaire la plus proche de 1. Si aucune variable fractionnaire n'est proche de 1, on augmente les coefficients de pénalité $\tau(i)$ afin de fixer davantage de variables aux itérations suivantes.

Comme dans l'heuristique H1, le solveur Xpress-MP est utilisé pour résoudre les PL initial et le MILP résiduel.

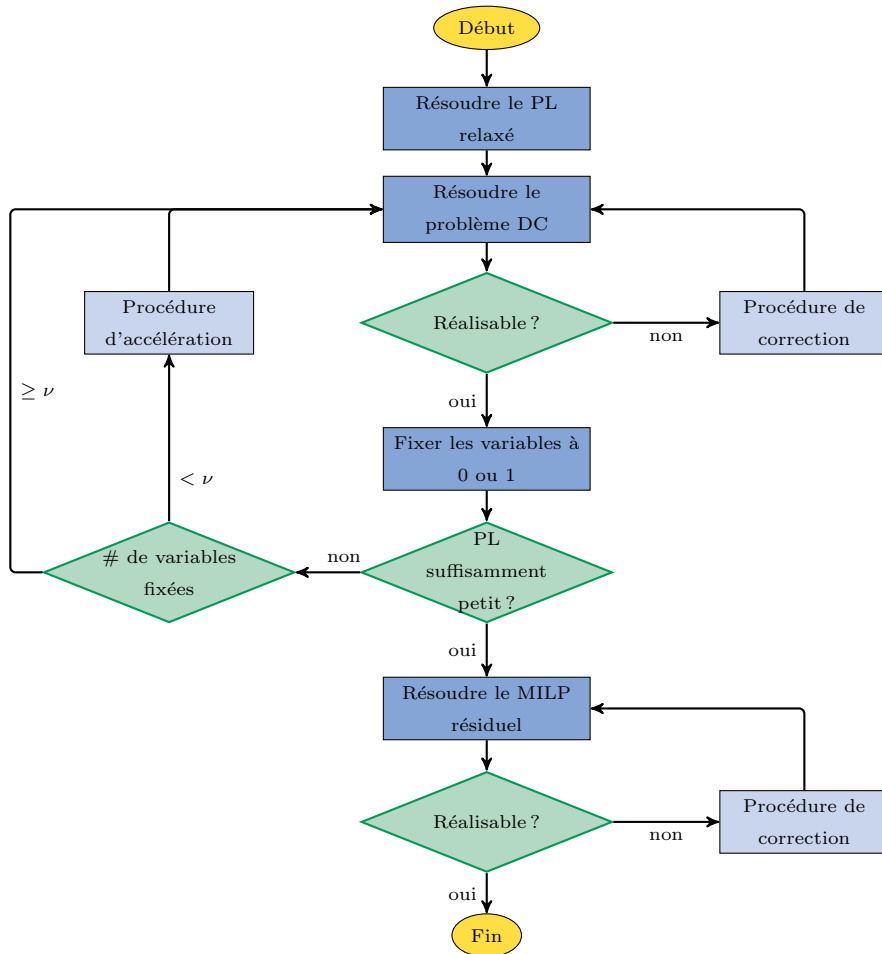


Figure 2.3 – Schéma général de l'heuristique H2

2.5 Heuristique H3 basée sur la relaxation lagrangienne

Une autre possibilité d'améliorer l'heuristique H1 est de renforcer le calcul de la borne inférieure, en utilisant la relaxation lagrangienne. Le principe de la relaxation lagrangienne est de relaxer ou une plusieurs contraintes jugées compliquantes, et de pénaliser leur violation dans la fonction objectif. Un facteur clé de réussite de la relaxation lagrangienne est le choix des contraintes à relaxer. Relaxer des contraintes couplantes, c'est-à-dire liant entre elles plusieurs variables du modèle, permet d'obtenir plusieurs sous-problèmes indépendants, et donc plus faciles à résoudre. Le choix des contraintes à relaxer diffère évidemment pour chaque problème d'optimisation. Dans cette section, nous détaillons la modélisation du problème lagrangien et sa décomposition en trois sous-problèmes notés *LR1*, *LR2* et *LR3*.

En recourant à la relaxation lagrangienne, deux améliorations par rapport à l'heuristique H1 sont attendues. D'une part, la borne inférieure obtenue avec la relaxation lagrangienne est toujours au moins aussi bonne que celle obtenue avec la relaxation linéaire (voir par ex. Guignard (2003)). D'autre part, le manque de mémoire du solveur pour résoudre les instances de grande taille montre la nécessité de décomposer (\mathcal{P}) en sous-problèmes indépendants.

2.5.1 Décomposition du problème

Le schéma de décomposition

Dans le modèle (\mathcal{P}), Les contraintes de satisfaction des demandes imposent des conditions sur les flux de matière en chaque entité. Si ces contraintes sont relaxées, le réseau peut être décomposé par échelon, c'est-à-dire en plusieurs sous-réseaux dont chacun contient un type d'entités et ses successeurs.

Dans des modèles SCND multi-échelons très hiérarchisés, les flux de matières vont uniquement de l'amont vers l'aval. Chaque site reçoit des produits depuis des sites situés à l'échelon amont, et les expédie vers des sites situés à l'échelon aval, comme illustré par la Figure 2.4.

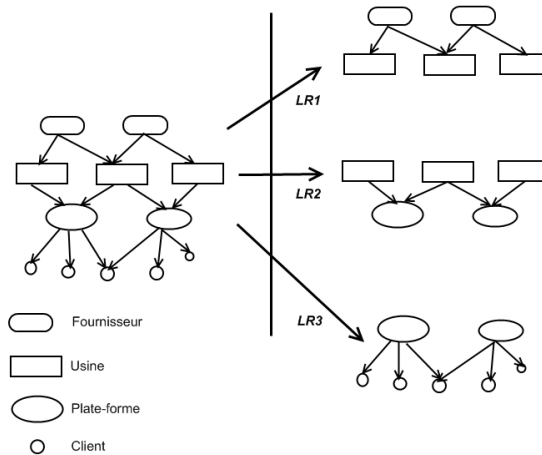


Figure 2.4 – Décomposition classique en échelons

En relaxant les contraintes d'égalité entre flux entrant et flux sortant à chaque échelon, il est alors possible de faire une décomposition parfaite par échelon (Hinojosa *et al.*, 2008). On obtiendrait alors trois sous problèmes à deux niveaux : fournisseurs/usines, usines/entrepôts et entrepôts/clients.

Le modèle (\mathcal{P}) incorpore quant à lui un recours à la sous-traitance, des flux de produits entre usines et des livraisons directes des usines vers les clients. Une décomposition classique par échelon n'est donc pas possible, ou du moins doit être adaptée. Nous définissons donc trois sous-problèmes, illustrés par la Figure 2.5.

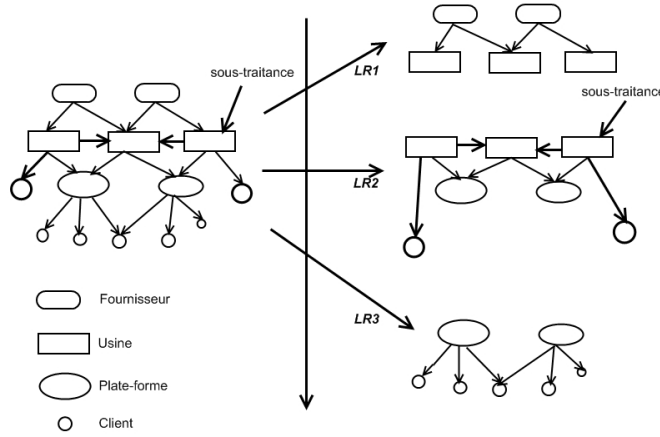


Figure 2.5 – Décomposition en échelons pour le modèle \mathcal{P}

Le sous-problème $LR1$ concerne l'échelon fournisseurs/usines. Le sous-problème $LR2$ concerne l'échelon usines / entrepôts et clients. Le sous-problème $LR3$ concerne l'échelon entrepôts/clients. Les trois sous-problèmes ainsi définis peuvent être à leur tour décomposés. Parmi l'ensemble des variables de décision, ce sont les variables de localisation x_i^t qui rendent le problème difficile à résoudre. On cherche donc à construire des sous-problèmes pour lesquels chacune de valeurs possibles de ces variables serait contenue dans un sous-problème distinct.

Comme x_i^t représente le statut de chaque site à chaque période, une décomposition par site et par période est nécessaire pour séparer ces variables. Les contraintes couplant ces variables entre elles sont les contraintes (2.1)–(2.3) du modèle (\mathcal{P}). Nous procédons donc aux décompositions par site et par période pour chaque sous-problème $LR1$, $LR2$, $LR3$. La Figure 2.6 résume ce schéma de décomposition.

1. **Problème $LR1$** : Il n'existe pas de contrainte sur le changement de statut des fournisseurs. Une décomposition par fournisseur et par période ne pose aucune difficulté. Chaque variable x_s^t est associée à un sous-problème, nommé $LR1_{s,t}$. Pour chaque couple $(s; t)$, x_s^t peut prendre soit la valeur 1 (le fournisseur s est sélectionné à la période t) soit la valeur 0 (s ne l'est pas). Deux scénarios sont associés à $LR1_{s,t}$ correspondant aux deux valeurs possibles de x_s^t . Une résolution par inspection de chaque sous problème $LR1_{s,t}$ est donc évidente.
2. **Problèmes $LR2$ et $LR3$** : Les usines et les entrepôts ont les mêmes types de contraintes d'ouverture et de fermeture. Le modèle empêche les sites privés de changer leur statut plus d'une fois durant l'horizon de temps. Une décomposition période par période comme pour $LR1$ est impossible. En revanche, on peut décomposer selon les périodes d'ouverture et de fermeture des sites. Pour les sites initialement ouverts, la période d'ouverture est connue ($t = 1$). Le modèle (\mathcal{P}) sert à déterminer une éventuelle période de fermeture. De même, les sites qui ouvrent pendant l'horizon de planification resteront ouverts jusqu'à la fin de l'horizon de temps. Nous associons à chaque usine i un couple (t_o, t_f) , où t_o représente la période d'ouverture et t_f représente la période de fermeture. Ces trois éléments définissent un sous-problème $LR2_{i,t_o,t_f}$. De même, nous définissons pour chaque entrepôt des sous-problèmes $LR3_{j,t_o,t_f}$. Les variables x_i^t sont fixées à 1 pour les $t \in [t_o, t_f]$ et à 0 sinon.

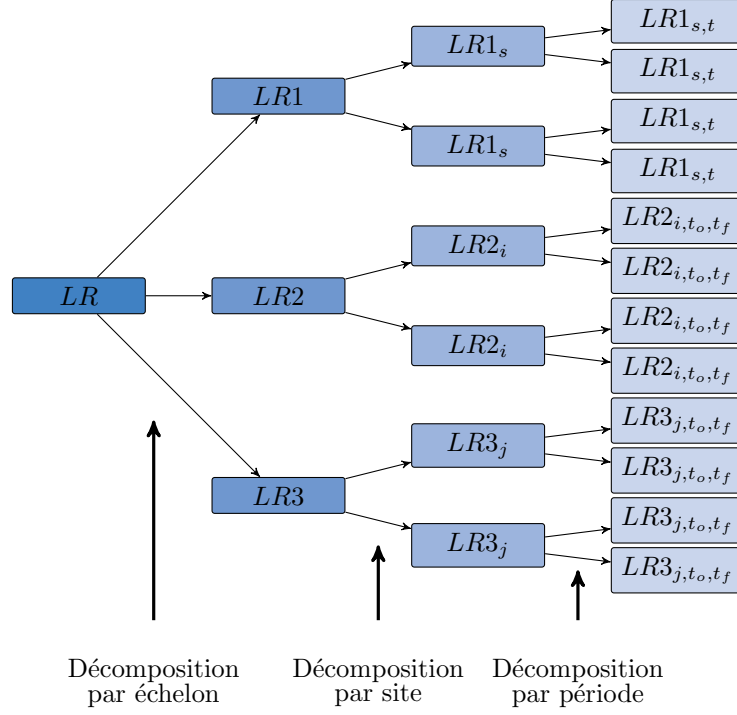


Figure 2.6 – Schéma de décomposition

Chaque sous-problème $LR2_{i,t_o,t_f}$ et $LR3_{j,t_o,t_f}$ peut être résolu par inspection. Leur modélisation complète ainsi que les méthodes de résolution sont détaillées dans la thèse de Phuong Nga Thanh (Thanh, 2008) et dans le rapport de recherche (Thanh *et al.*, 2011).

Un schéma semblable est utilisé par Hinojosa *et al.* (2008) pour un problème comparable, mais ne comportant ni sélection de fournisseurs ni échange de produits entre usines.

Relaxation partielle des contraintes (2.12)

Parmi les contraintes de capacité, il existe un groupe couplant non seulement sur les produits mais aussi sur les échelons. Ce sont les contraintes (2.12).

$$\sum_{p \in P_f} WS_{p,j} \cdot \left(q_4^t(p,j) + \sum_{i \in \Gamma^-(j)} \frac{q_1^t(p,i,j)}{\Delta_{i,j}} \right) \leq \bar{U}_j^t (KS_j x_j^t + \sum_{o \in O} KS_o y_{j,o}^t) \quad \forall t \in T, j \in J.$$

Ces contraintes relient des flux $q_1^t(p,i,j)$ de l'échelon usines/entrepôts et les variables x_j^t et $y_{j,o}^t$ concernant l'état des entrepôts de l'échelon suivant. Pour une décomposition par échelon, ces contraintes nécessitent d'être relaxées. Néanmoins, ce sont des contraintes de capacité que nous souhaitons ne pas relaxer complètement. Pour trouver un compromis, nous décidons de relaxer les contraintes (2.12) et d'ajouter les contraintes (12bis) reliant seulement les stocks et la capacité de stockage :

$$\sum_{p \in P_f} WS_{p,j} \cdot q_4^t(p,j) \leq \bar{U}_j^t \cdot \left(KS_j \cdot x_j^t + \sum_{o \in O} KS_o \cdot y_{j,o}^t \right) \quad \forall t \in T, \forall j \in J \quad (12bis)$$

Les contraintes (12bis) ne sont pas couplantes et peuvent être insérées dans le sous-problème $LR3$.

Formulation des sous-problèmes

Aux quatre groupes de contraintes relaxés (2.1),(2.2),(2.3) et (2.12) sont associés les multiplicateurs respectifs $\lambda_{c,p}^t$, $\beta_{j,p}^t$, $\mu_{i,p'}^t$ et γ_j^t avec $\lambda_{c,p}^t$, $\mu_{i,p'}^t$, $\gamma_j^t \geq 0$ et $\beta_{j,p}^t$ quelconque.

La relaxation lagrangienne $LR(\lambda, \beta, \mu, \gamma)$ s'écrit donc de la façon suivante :

$$\begin{aligned}
\text{Min } z &+ \sum_{t \in T} \sum_{c \in C} \sum_{p \in P_f} \lambda_{c,p}^t \left(D_{c,p}^t - \sum_{i \in \Gamma_c^-} q_1^t(p, i, c) \right) \\
&+ \sum_{t \in T} \sum_{j \in J} \sum_{p \in P_f} \beta_{j,p}^t \left(\sum_{c \in \Gamma_j^+} q_1^t(p, j, c) + q_4^t(p, j) - q_4^{t-1}(p, j) - \sum_{i \in \Gamma_j^-} q_1^t(p, i, j) \right) \\
&+ \sum_{t \in T} \sum_{i \in I} \sum_{p' \in P_r} \mu_{i,p'}^t \left(\sum_{p \in P_m} \text{BoM}_{p',p} \cdot q_2^t(p, i) - \sum_{s \in \Gamma_i^-} q_1^t(p, s, i) \right) \\
&+ \sum_{t \in T} \sum_{j \in J} \gamma_j^t \left(\sum_{p \in P_f} \sum_{i \in \Gamma_j^-} W_{S_{p,j}} \frac{1}{L_{i,j}} \cdot q_1^t(p, i, j) + \sum_{p \in P_f} W_{S_{p,j}} \cdot q_4^t(p, j) - \bar{U}_j^t \cdot (KS_j \cdot x_j^t + \sum_{o \in O} KS_o \cdot y_{j,o}^t) \right)
\end{aligned}$$

s.t. (2.4) – (2.34).

Selon les flux de matière considérés, et en appliquant le schéma de décomposition illustré par la Figure 2.5, les termes de la fonction objectif peuvent être répartis dans les trois-sous-problèmes :

$$\begin{aligned}
&\sum_{t \in T} \sum_{c \in C} \sum_{p \in P_f} \lambda_{c,p}^t D_{c,p}^t && \rightarrow \text{Composante indépendante} \\
&-\sum_{t \in T} \sum_{c \in C} \sum_{p \in P_f} \lambda_{c,p}^t \sum_{i \in \Gamma_c^-} q_{p,i,c}^t && \rightarrow \text{Problème LR2} \\
&-\sum_{t \in T} \sum_{c \in C} \sum_{p \in P_f} \lambda_{c,p}^t \sum_{j \in \Gamma_c^-} q_{p,j,c}^t && \rightarrow \text{Problème LR3} \\
&\sum_{t \in T} \sum_{j \in J} \sum_{p \in P_f} \beta_{j,p}^t \left(\sum_{c \in \Gamma_j^+} q_1^t(p, j, c) + q_4^t(p, j) - q_4^{t-1}(p, j) \right) && \rightarrow \text{Problème LR3} \\
&-\sum_{t \in T} \sum_{j \in J} \sum_{p \in P_f} \beta_{j,p}^t \sum_{i \in \Gamma_j^-} q_1^t(p, i, j) && \rightarrow \text{Problème LR2} \\
&\sum_{t \in T} \sum_{i \in I} \sum_{p' \in P_r} \mu_{i,p'}^t \sum_{p \in P_m} \text{BoM}_{p',p} \cdot q_2^t(p, i) && \rightarrow \text{Problème LR2} \\
&-\sum_{t \in T} \sum_{i \in I} \sum_{p' \in P_r} \mu_{i,p'}^t \sum_{s \in \Gamma_i^-} q_1^t(p', g, i) && \rightarrow \text{Problème LR1} \\
&\sum_{t \in T} \sum_{j \in J} \gamma_j^t \sum_{p \in P_f} \sum_{i \in \Gamma_j^-} W_{S_{p,j}} \frac{1}{L_{i,j}} \cdot q_1^t(p, i, j) && \rightarrow \text{Problème LR2} \\
&\sum_{t \in T} \sum_{j \in J} \gamma_j^t \left(\sum_{p \in P_f} W_{S_{p,j}} \cdot q_4^t(p, j) - \bar{U}_j^t \cdot (KS_j \cdot x_j^t + \sum_{o \in O} KS_o \cdot y_{j,o}^t) \right) && \rightarrow \text{Problème LR3}
\end{aligned}$$

La proposition 2.1 exprime la relation entre la valeur optimale du problème $LR(\lambda, \beta, \mu, \gamma)$ et celle des trois sous-problèmes $LR1$, $LR2$ et $LR3$.

Proposition 2.1. Soit $Z(\cdot)$ la valeur de la fonction objectif du problème (\cdot) alors :

$$Z(LR(\lambda, \beta, \mu, \gamma)) = Z(LR1(\mu)) + Z(LR2(\lambda, \beta, \mu, \gamma)) + Z(LR3(\lambda, \beta, \gamma)) + \sum_{t \in T} \sum_{p \in P_f} \sum_{c \in C} \lambda_{c,p}^t \cdot D_{c,p}^t$$

Les composantes de la fonction objectif initiale ainsi que les pénalités sont toutes associées à l'un des trois sous-problèmes. Il reste uniquement un terme $\sum_{t,c,p} \lambda_{c,p}^t \cdot D_{c,p}^t$ indépendant. D'autre part, les trois sous-problèmes sont indépendants les uns des autres. Ainsi minimiser la fonction objectif du problème lagrangien revient à minimiser chacun des trois sous-problèmes, ce qui est possible par inspection.

2.5.2 Schéma général de l'heuristique H3

L'heuristique H3 de relaxation lagrangienne suit un schéma assez classique, décrit par le logigramme de la Figure 2.7.

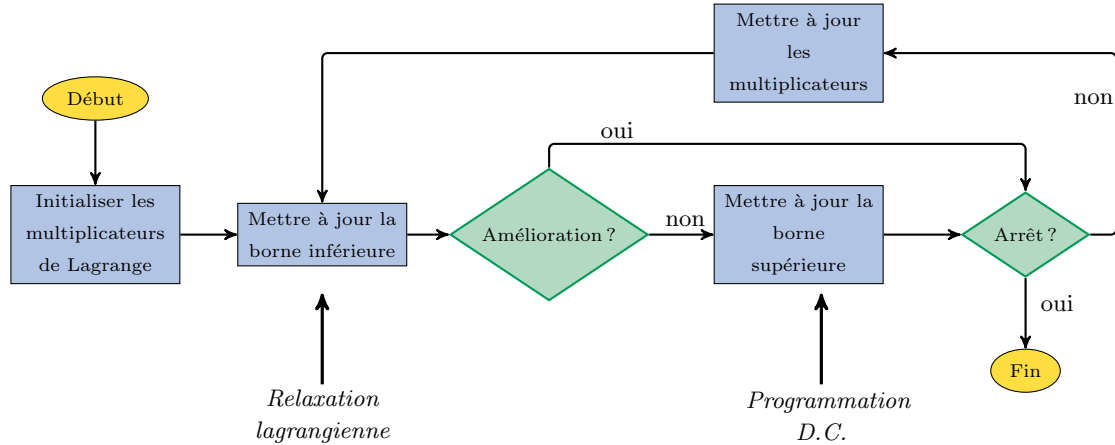


Figure 2.7 – Schéma général de l'heuristique H3

La résolution des sous-problèmes $LR1(\mu)$, $LR2(\lambda, \beta, \mu, \gamma)$, $LR3(\lambda, \beta, \gamma)$ permet de calculer une **borne inférieure** à chaque itération. Un point original est qu'on utilise l'algorithme DCA comme heuristique lagrangienne. Le Thi *et al.* (1999) montrent en effet que pour les problèmes MILP, si la solution initiale est entière et le coefficient des fonctions de pénalité τ suffisamment grand, la solution obtenue à chaque itération de DCA reste entière. DCA est initialisé avec une solution construite à partir de la résolution de $LR1$, $LR2$ et $LR3$. Cette solution est le plus souvent non réalisable, ce qui ne dérange pas le fonctionnement de DCA. Lorsque la solution produite par DCA est réalisable, elle est une bonne borne supérieure de la solution optimale de (P) . Comme l'algorithme DCA est une méthode de recherche locale, deux solutions initiales très proches aboutiront vraisemblablement au même optimum local. L'heuristique H3 ne lance donc l'algorithme DCA que lorsque la borne inférieure a évolué de manière significative depuis le dernier lancement.

Les multiplicateurs sont calculés par une méthode de sous-gradient. Comme proposé par Hinojosa *et al.* (2008), ils sont initialisés avec les valeurs duales associées aux contraintes relaxées. Nous utilisons deux critères d'arrêt. Le premier se base sur l'écart relatif entre la borne supérieure et la borne inférieure. Le deuxième se base sur le nombre maximal d'itérations.

2.6 Evaluation des trois heuristiques

Les trois heuristiques ont été testées sur un jeu de 450 instances générées aléatoirement. Toutes les instances sont construites avec 5 périodes de temps et 3 options de capacité pour chaque site. Les plus petites d'entre elles comportent 30 sites candidats, 100 clients et 10 produits. Les plus grandes ont 62 sites candidats, 270 clients et 18 produits. Le nombre de variables binaires dans le modèle (\mathcal{P}) va ainsi de 750 à 2185, dont 300 à 750 variables de localisation.

Les 450 instances sont organisées en 15 familles de 30 instances de même taille, mais avec des niveaux de demande et des configuration logistiques distinctes. Les familles d'instances sont nommées S1–S5 (small), M1–M5 (medium), L1–L5 (large). Les principes suivis pour la génération des instances sont détaillés dans Thanh *et al.* (2008).

On compare les résultats des heuristiques avec un résultat dit de référence obtenus avec le solveur Xpress-MP avec un temps de calcul limité à 3 heures. Le solveur est capable de résoudre la totalité des instances des familles S1 à S5 dans un temps de calcul compris entre 65 et 1419 secondes. En revanche, il ne trouve des solutions optimales que pour 29 instances M1 sur 30, 2 instances M5 sur 30 et seulement une seule instance sur les 150 des familles L1 à L5.

Nous évaluons les trois heuristiques sur deux critères principaux : le temps d'exécution et la valeur de la fonction objectif. On note Δ_z l'écart relatif entre une solution de référence et une solution heuristique. De même Δ_t exprime l'écart relatif entre les temps de calcul. Lorsqu'une instance n'est pas résolue optimalement par le solveur, la valeur de Δ_z peut donc être négative. Le Tableau 2.1 résume les valeurs de Δ_z et Δ_t pour les trois heuristiques.

Familles	Δ_z (en %)			Δ_t (en %)		
	H1	H2	H3	H1	H2	H3
S1	1.80	1.10	0.36	-57	-41	116
S2	2.20	1.20	0.19	-37	-26	120
S3	1.68	0.50	0.09	-50	-37	46
S4	1.19	0.77	0.18	-82	-70	5.1
S5	0.53	0.38	0.16	-72	-60	-19
M1	1.72	0.45	0.38	-65	-65	-23
M2	1.15	0.81	0.4	-81	-83	-54
M3	0.85	0.2	0.31	-77	-80	-35
M4	1.86	1.48	0.2	-80	-85	-56
M5	2.07	1.16	0.27	-83	-86	-49
L1	2.49	1.71	0.51	-83	-92	-65
L2	1.09	0.70	0.18	-72	-86	-63
L3	1.62	0.76	0.67	-71	-84	-61
L4	1.56	0.65	0.53	-69	-82	-55
L5	2.88	-0.69	-1.20	-42	-83	-51

Tableau 2.1 – Comparaison des trois heuristiques : fonction objectif et temps d'exécution

La relaxation lagrangienne obtient clairement de meilleurs résultats en terme de fonction objectif. Lorsque le solveur est capable d'obtenir une solution optimale, cette méthode obtient un saut d'optimalité généralement inférieur à 0.5%. Pour les instances de plus grande taille, les résultats sont plus difficiles à interpréter car l'écart relatif entre les bornes inférieure et supérieure fourni par le solveur peut monter à 4%. Toujours est il que sur les instances de la famille L5, la relaxation lagrangienne permet d'obtenir des résultats meilleurs que le solveur en un temps deux fois plus court.

L'heuristique H1 est la plus rapide pour les instances de petite taille, car la relaxation linéaire peut alors être résolue très rapidement pour ces instances. Ensuite, DCA est clairement plus rapide. La relaxation lagrangienne nécessite un temps de calcul plus long pour presque toutes les familles, mais l'écart avec les deux autres heuristiques se réduit pour les instances les plus grandes.

Afin d'avoir une vision plus complète des résultats numériques, nous avons représenté graphiquement les valeurs de Δ_z et Δ_t pour chacune des heuristiques. Chaque point des Figures 2.8 à 2.10 représente une des 150 instances des familles L1 à L5. L'axe des abscisses représente la valeur Δ_t et l'axe des ordonnées Δ_z .

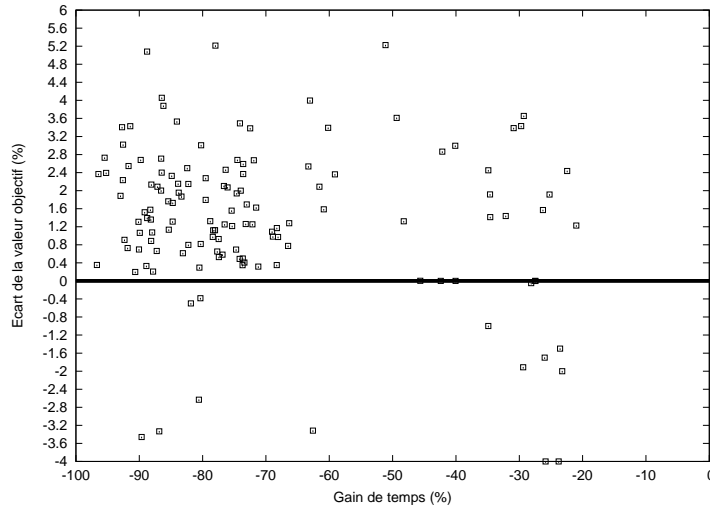


Figure 2.8 – Heuristique H1 : familles L1 à L5

Les résultats de l'heuristique H1 montrent une majorité de points au dessus de l'axe des abscisses, c'est-à-dire moins bons que le solveur. La plupart des résultats montrent un écart de 1% et 3% avec le solveur, mais on observe une grande dispersion des résultats. Cela vaut pour la valeur de la fonction objectif comme pour le temps de calcul.

Le nombre des points au dessus de l'axe des abscisses diminue avec l'heuristique H2, et encore plus avec l'heuristique H3, ce qui montre l'amélioration de la qualité de la fonction objectif avec ces méthodes. Dans la Figure 2.9, la plupart des points se situent à moins de 2% de la valeur de référence, pour une réduction de temps de calcul de plus de 60%. Nous observons cependant quelques instances pour lesquelles les résultats sont moins bons. L'heuristique H3, bien que moins rapide, apparaît comme plus stable en terme de valeur de la fonction objectif. Toutes les instances sont résolues avec un écart relatif de moins de 2% par rapport au solveur. De plus cette méthode semble mieux supporter mieux le passage à l'échelle.

2.7 Conclusions et pistes de recherche

Dans ce chapitre, nous avons décrit trois heuristiques pour la résolution d'un modèle riche de conception de réseau logistique. Le modèle, publié en 2008 (Thanh *et al.*, 2008), cumule plusieurs contraintes réalistes comme la prise en compte de la sous-traitance, l'utilisation temporaire d'entrepôts, la modélisation des phases de démarrage et d'arrêt de l'activité, et la prise en compte

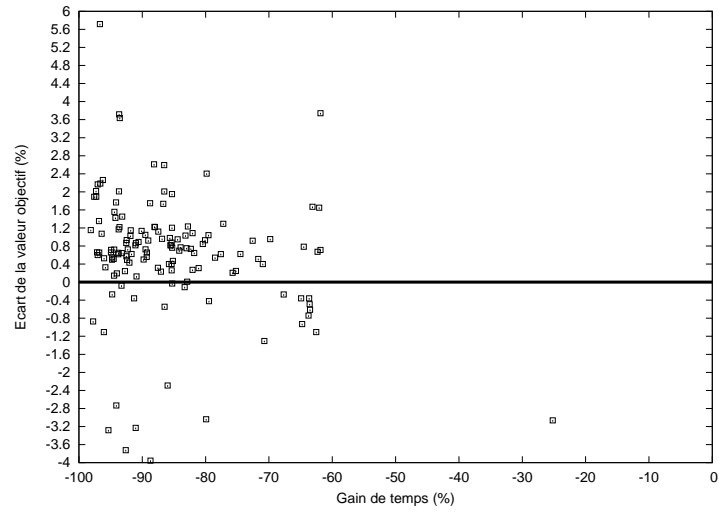


Figure 2.9 – Heuristique H2 : familles L1 à L5

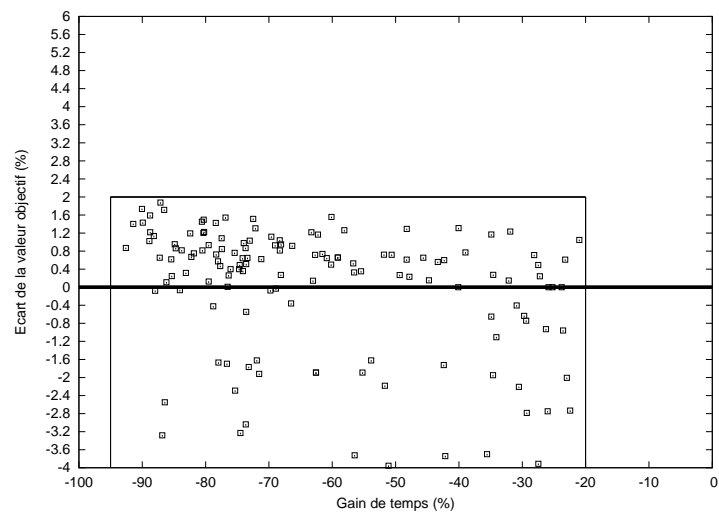


Figure 2.10 – Heuristique H3 : familles L1 à L5

d'un taux minimal et maximal d'utilisation des sites. De plus, la génération de 450 instances de tests a permis la comparaison des heuristiques sur une base élargie d'instances.

Les trois heuristiques de résolution décrites dans ce chapitre s'appuient sur des techniques éprouvées de recherche opérationnelle. L'introduction de l'algorithme DCA comme brique de résolution incluse dans une heuristique plus large (heuristique H2) ou comme heuristique lagrangienne est un point original de nos travaux, qui a montré son efficacité en terme de fonction objectif et de temps de calcul. Plus généralement, les résultats numériques montrent la rapidité de l'heuristique H2 et la stabilité des résultats de l'heuristique H3.

Notre contribution réside dans l'adjonction de procédures spécifiques pour la sélection de variables clés, l'accélération de certaines phases algorithmiques et l'introduction de règles d'arrondi flexibles. Ces procédures sont généralement liées à notre vision des modèles de type SCND, centrés sur les variables de localisation. Les trois heuristiques suivent donc le même principe : identifier les attributs qui compliquent la résolution du modèle (variables de localisation, contraintes couplantes, etc.) et les traiter de manière prioritaire. Ces idées ont également été appliquées dans d'autres travaux récents qui visent à résoudre de façon exacte ou quasi-exacte des problèmes de localisation de grande taille. L'idée générale de l'heuristique H1 d'identifier et d'exploiter un sous-ensemble de variables clés se retrouve dans la notion *kernel search* (Guastaroba et Speranza, 2012). Avella *et al.* (2009) utilisent la relaxation Lagrangienne pour sélectionner un sous-ensemble de variables définissant un *core problem*, lequel est résolu avec un algorithme de type branch-and-cut. Ces approches permettent de résoudre des instances de grande taille du Capacitated Facility Location Problem. L'adaptation de ces approches pour résoudre des modèles riches de conception de chaîne logistique est une piste de recherche intéressante.

Concernant les méthodes purement heuristiques, une amélioration des heuristiques H1 à H3 est bien sûr possible, mais au prix d'une analyse très fine du comportement des algorithmes, et sans doute d'un gros effort de paramétrage. On voit donc les limites pratiques de telles approches.

Parmi les multiples méthodes heuristiques utilisées dans la littérature pour résoudre les modèles de type SCND, les heuristiques ou métaheuristiques basées sur des recherche locales occupent finalement assez peu de place. Cela peut s'expliquer par la difficulté de parcourir le voisinage d'une solution donnée. La modification d'une variable binaire (ouverture d'un site, choix d'une technologie de fabrication ou d'un mode de transport) en un point du réseau est en effet susceptible de modifier les flux dans l'ensemble du réseau. On peut opposer des méthodes de recherche locale qui ne modifient qu'un tout petit nombre de variables de la solution courante à chaque itération (avec l'espoir que l'impact de cette modification soit limité) aux méthodes à voisinage large qui fixent a priori quelle part de la solution peut être modifiée. La combinaison de ces deux approches me paraît être une piste de recherche intéressante. Les facteurs clés de succès seraient une implémentation incrémentale des voisinages (pour éviter de tout recalculer à chaque itération) et l'auto-paramétrage des heuristiques. Enfin, étant donné la multiplicité des modèles de conception de réseau, les méthodes de résolution se doivent d'être suffisamment génériques pour pouvoir s'adapter à des nombreuses variantes.

Chapitre 3

Transports à la demande

3.1 Introduction

Le problème de tournées de véhicules (VRP : Vehicle Routing Problem) est l'un des plus emblématiques de l'optimisation combinatoire, tant il a suscité d'efforts de recherche et d'algorithmes innovants. La formulation classique du VRP consiste à confectionner des tournées de véhicules visitant un ensemble de sites clients à partir d'un dépôt, de manière à minimiser un coût lié au nombre de véhicules utilisés ou à la distance totale parcourue.

Ce problème a fait l'objet de nombreuses extensions : problème de tournées de véhicules avec fenêtres horaires, tournées avec profit, tournées sur les arcs, tournées avec flotte hétérogène, inventory routing problem, pickup and delivery problem (PDP), dial-a-ride problem (DARP) etc. Nous renvoyons aux ouvrages de Toth et Vigo (2002) et Golden *et al.* (2008) pour une présentation plus complète du VRP et de ses variantes.

Dans ce chapitre, nous nous intéressons plus particulièrement au problème de tournées de véhicules appliqué aux transports de personnes handicapées. Les techniques de modélisation et les méthodes de résolution diffèrent peu qu'il s'agisse de personnes ou de marchandises. Il est en revanche évident qu'on ne transporte pas des personnes comme on transporte des colis ou des palettes. La notion de qualité de service est à prendre en compte, à plus forte raison en présence d'un public sensible.

La section 3.2 de ce chapitre retrace le projet Marika (2007-2008), qui a consisté à concevoir un logiciel d'aide à la décision pour l'ADAPEI 44 (Association Départementale des Amis et Parents de Personnes Handicapées Mentales). Le logiciel s'adressait à une vingtaine d'établissements de type IME (Institut Médico-Educatif) ou ESAT (Etablissements et Services d'Aides par le Travail), accueillant des enfants ou adultes à la journée. En raison de leur manque d'autonomie, ces personnes ne peuvent généralement pas se rendre à leur établissement par leurs propres moyens. Les établissements sont donc chargés d'organiser les transports domicile-établissement, ce qui représente le troisième poste de dépenses après les salaires et la restauration : environ 2500€ par an et par personne en moyenne (Caisse Nationale de Solidarité pour l'Autonomie, 2009).

Même si le projet Marika n'a donné lieu qu'à des publications en conférence (Lehuédé *et al.*, 2008; Péton *et al.*, 2008; Lehuédé *et al.*, 2009b) ou en revue nationale (Lehuédé *et al.*, 2009a), ce projet a permis d'identifier plusieurs pistes de recherche :

- **Régularité des horaires de service** : comment proposer des horaires de service réguliers tout au long de la semaine alors que la demande peut varier d'un jour sur l'autre ? La section 3.3 est consacrée à cette question.
- **Mutualisation et optimisation multi-critères** : comment proposer des transports mutualisés entre plusieurs établissements, conciliant plusieurs objectifs parfois antagonistes ? La section

3.4 est consacrée à cette question.

- **Tournées avec transferts** : Comment construire un service de transport multi-établissement autorisant le transfert de passagers d'un véhicule à un autre ? Ce travail, qui a fait l'objet d'une thèse de doctorat, sera développé dans le chapitre 4.

3.2 Projet Marika : une recherche tabou pour la résolution d'un open-VRPTW

3.2.1 Description du problème et modélisation

Nous nous plaçons dans le cas d'un établissement qui doit organiser un service de transport chaque matin et chaque après-midi de la semaine. Les tournées sont effectuées par une flotte de véhicules appartenant pour l'essentiel à des sociétés de transport (taxis de petite ou grande capacité, minibus, bus). Ces véhicules effectuent donc des tournées ouvertes. La définition d'une tournée est très variable selon les transporteurs. Le départ d'une tournée est soit le domicile du chauffeur soit la première adresse desservie. Les établissements peuvent également posséder des véhicules utilisés en appoint pour effectuer les tournées.

Chaque véhicule possède une capacité en places assises, et certains d'entre eux une capacité en fauteuils roulants. Chaque passager peut avoir des contraintes personnelles qui imposent des fenêtres horaires. Un véhicule a le droit d'arriver sur un lieu de prise en charge en avance par rapport à une fenêtre horaire. Dans ce cas, il attend sur place l'ouverture de la fenêtre horaire. Le temps de transport par jour et par personne est limité par la législation. En outre, certaines limitations individuelles de temps de transport peuvent être ajoutées pour raisons médicales. De nombreuses personnes peuvent avoir plusieurs adresses au long de la semaine (adresse principale, famille d'accueil, foyer, nourrice, grands-parents, etc.). Mais les demi-journées sont traitées indépendamment les unes des autres, cela ne porte pas à conséquence.

Nous avons retenu deux critères d'évaluation des tournées :

- **Le coût** : les coûts de transport peuvent être très variables selon le type de contrat de transport. Il peut dépendre du nombre de véhicules utilisés et de leur taille, du nombre de kilomètres parcourus, du temps de transport, etc. En pratique, il est très difficile de connaître le mode de calcul du coût. En l'absence de données précises, il est donc illusoire de vouloir minimiser directement le coût du transport. En revanche, le temps de parcours des véhicules est un indicateur de coût reconnu comme pertinent et valable à la fois en milieu rural et urbain.
- **La qualité de service aux usagers** : il existe plusieurs métriques pour mesurer la qualité de service en terme de temps de transport. Dans un contexte de transport par bus scolaires, Spada *et al.* (2005) minimisent le détour moyen des personnes. Ce détour est la différence entre la durée d'un trajet direct vers l'établissement et le temps de transport de cette personne dans la tournée. Minimiser ce critère revient à rechercher une efficacité globale du système, au détriment possible de certains individus. Une autre approche est de minimiser le détour maximal (sur l'ensemble des usagers). Cela a pour effet d'obtenir un planning plus équitable, mais aussi plus coûteux. Paquette *et al.* (2009) donnent une description détaillée des différentes fonctions possibles et leurs conséquences.

En conclusion, le problème traité peut se modéliser comme un VRP avec fenêtres horaires et quelques contraintes ou caractéristiques additionnelles :

- la flotte est hétérogène ;
- il peut être demandé d'utiliser certains véhicules en priorité (par exemple ceux de l'établissement) ;
- présence de tournées ouvertes et de tournées fermées ;
- contrainte de limitation de temps de transport pour les passagers ;
- contraintes supplémentaires d'affectation passagers-véhicules (spécificités des personnes en fauteuil roulant) ou d'incompatibilités entre passagers ;

3.2.2 Positionnement dans la littérature

Le VRP avec tournées ouvertes, ou open-VRP n'avait donné lieu qu'à très peu de travaux jusqu'en 2004. Brandão (2004) ne mentionne en effet que l'existence de trois articles antérieurs. Le problème connaît depuis lors un vif regain d'intérêt. Diverses méthodes ont été employées : algorithmes génétiques (Rekiek *et al.*, 2006), méthode record-to-record (Li *et al.*, 2007), colonies de fourmis (Li et Tian, 2006) ou recherche à voisinage variable (Fleszar *et al.*, 2009).

La plupart des articles considèrent des versions de l'open VRP sans fenêtres horaires et avec des flottes de véhicules homogènes. Fu *et al.* (2005) intègrent la notion de flotte hétérogène. Repoussis *et al.* (2007) décrivent un modèle d'open-VRP avec fenêtres horaires. Pisinger et Ropke (2007) applique leur recherche à voisinage large à l'open-VRP avec fenêtres horaires. Enfin, Liu et Jiang (2012) étudient le VRP avec à la fois des routes ouvertes et fermées. Concernant les méthodes de résolution, seuls Letchford *et al.* (2007) proposent une résolution exacte de l'open-VRP. Le temps de résolution peut dépasser une heure pour des instances de quelques dizaines de sites.

Notre choix de méthode de résolution a été guidé par des considérations pratiques. Il nous fallait produire des routes de bonne qualité en un temps de calcul limité et avec un temps de développement restreint. Nous avons donc opté pour une méthode de recherche tabou, qui a largement fait ses preuves pour des problèmes similaires (Cordeau et Laporte, 2005).

3.2.3 Recherche tabou pour le problème de transport de personnes handicapées

Nous avons proposé une méthode de recherche tabou reposant sur quelques voisinages classiques en tournées de véhicules :

- **déplacement d'une personne** : une personne est retirée de sa tournée et insérée à un autre emplacement, dans la même tournée ou non,
- **échange de deux personnes** : deux personnes, de la même tournée ou non, échangent leurs positions,
- **cross exchange** : sélectionner deux personnes a et b sur deux tournées différentes, et échanger les fins de tournées après a et b ,
- **string exchange** : échange de deux chaînes de plusieurs personnes consécutives (nous nous limitons à des échanges de chaînes d'au plus deux personnes),
- **chaînes d'éjection** : effectuer simultanément plusieurs déplacements de personnes (a remplace b , qui remplace c etc.), de manière à faciliter les mouvements dans des tournées déjà pleines. La complexité de ce mouvement étant supérieure à celle des autres voisinages, nous nous limitons à des chaînes d'éjection comportant deux mouvements. Un voisinage similaire est présenté par Toth et Vigo (1997).

Il existe plusieurs possibilités pour sélectionner les personnes à déplacer et les positions auxquelles les replacer dans les tournées. Une solution rapide consiste effectuer une sélection aléatoire de personnes à déplacer. Spada *et al.* (2005) utilisent une probabilité basée sur le détour individuel de chaque personne. L'approche retenue consiste à examiner l'ensemble du voisinage de la solution courante et à sélectionner la meilleure solution voisine. Malgré le temps de calcul plus long à chaque itération, cela s'est finalement avéré assez efficace. Une autre caractéristique notable de l'heuristique est l'alternance entre les différents voisinages. Contrairement à Aksen *et al.* (2007) ou Fu *et al.* (2005), nous évitons de procéder à un tirage au sort pour sélectionner le voisinage à utiliser. Nous souhaitons en effet obtenir un algorithme déterministe, de manière à assurer la reproductibilité d'une solution et offrir un comportement stable aux utilisateurs. Cet argument est également avancé par Brandão (2004). En pratique, la fréquence de recours à chaque voisinage a été déterminée expérimentalement. Une méthode basée sur l'apprentissage constituerait une amélioration de notre approche. Quel que soit le voisinage utilisé, nous concluons chaque itération par une amélioration locale de chaque tournée par des déplacements, échanges de personnes ou des opérations de type 2-opt.

Cet algorithme a été codé en C++ et intégré dans un logiciel livré à l'ADAPEI 44 en 2008. Sur les jeux de données fournis par l'ADAPEI44 pendant la durée de l'étude, nous avons observé un gain moyen de 14% sur la durée de tournées, gain moyen qui cache une très grande variabilité.

3.3 Projet cTouVeR : conception de Tournées de Véhicules Régulières

Les personnes handicapées mentales sont particulièrement sensibles aux perturbations de leur quotidien. Il est donc souhaitable que les personnes ayant une demande de transport identique tous les jours soient pris en charge à un horaire identique, et si possible par le même chauffeur. Même si le critère de minimisation du coût reste prédominant, il est souhaitable vis-à-vis des usagers particulièrement sensibles aux changements de proposer un service régulier.

La notion de régularité a été introduite par Groër *et al.* (2009) pour des services de messagerie : l'aptitude de tels services à proposer des heures de passages identiques chaque jour contribue à la satisfaction des principaux clients et est un avantage concurrentiel pour les compagnies qui y parviennent. Dans cette section, nous proposons notre propre modélisation de la notion de régularité. Nous présentons ensuite un modèle mathématique de tournées de véhicules multi-périodiques incluant la régularité. Ce problème est résolu par une recherche à voisinage large, nécessitant à chaque itération de résoudre un sous-problème inédit de tournées. Nous décrivons ce sous-problème et sa résolution par une méthode de génération de colonnes. Enfin, nous présentons des résultats expérimentaux contribuant à chiffrer le coût de la régularité dans les tournées de véhicules.

Ce travail a été réalisé dans le cadre du projet GdR RO cTouVeR (conception de Tournées de Véhicules Régulières) en 2008-2009. Il a été publié dans le journal *Networks* (Feillet *et al.*, 2014).

3.3.1 Modélisation de la régularité de service

Considérons un problème de tournées sur un ensemble de jours $D = \{1, \dots, |D|\}$ et un client dont les heures de service sont données par le vecteur $W = (w_1, \dots, w_{|D|})$.

Groër *et al.* (2009) expriment la régularité de service sur l'horizon de temps par l'écart

$$\max_{1 \leq d \leq |D|} w_d - \min_{1 \leq d \leq |D|} w_d,$$

appelé *slack time* entre les heures de passage extrêmes. Ils appliquent de plus un *principe de précedence* selon lequel si deux clients i_1 et i_2 sont servis dans cet ordre un jour donné, alors ils sont toujours servis dans cet ordre. Ce principe de précedence permet la construction de tournées type pour les clients réguliers. Les clients intermittents peuvent facilement s'insérer dans une tournée type.

L'approche avec un *slack time* et le principe de précedence est pertinente pour une activité de messagerie, car le prestataire sert de multiples clients chaque jour avec une fenêtre de temps assez large pour l'heure de service de chacun. Elle l'est moins pour un transport de personnes, où les fenêtres horaires sont beaucoup plus resserrées et où chaque tournée ne comporte que quelques clients. La notion de tournée type a une moindre importance.

Pour les applications de transport à la demande, nous supposons qu'une très légère variation des horaires d'un jour sur l'autre passe inaperçue. En revanche, une différence d'horaire sensible d'un jour sur l'autre crée un désagrément, mais ce désagrément ne varie pas avec la différence d'horaires. En d'autres termes, c'est l'existence de différents horaires dans la semaine, mais pas la valeur de ces horaires, qui diminue la qualité de service perçue.

Pour illustrer cette approche de la régularité, observons les cinq plannings présentés dans la Figure 3.1. Les plannings W_3 et W_4 , bien que possédant la même valeur de *slack time*, n'ont pas le même

profil. W_4 montre davantage de stabilité autour de deux horaires de service : 8h et 9h. D'autre part, les planning W_4 et W_5 , ont des valeurs de slack time très différentes, mais un client jugera probablement identique la qualité de service de ces deux plannings.

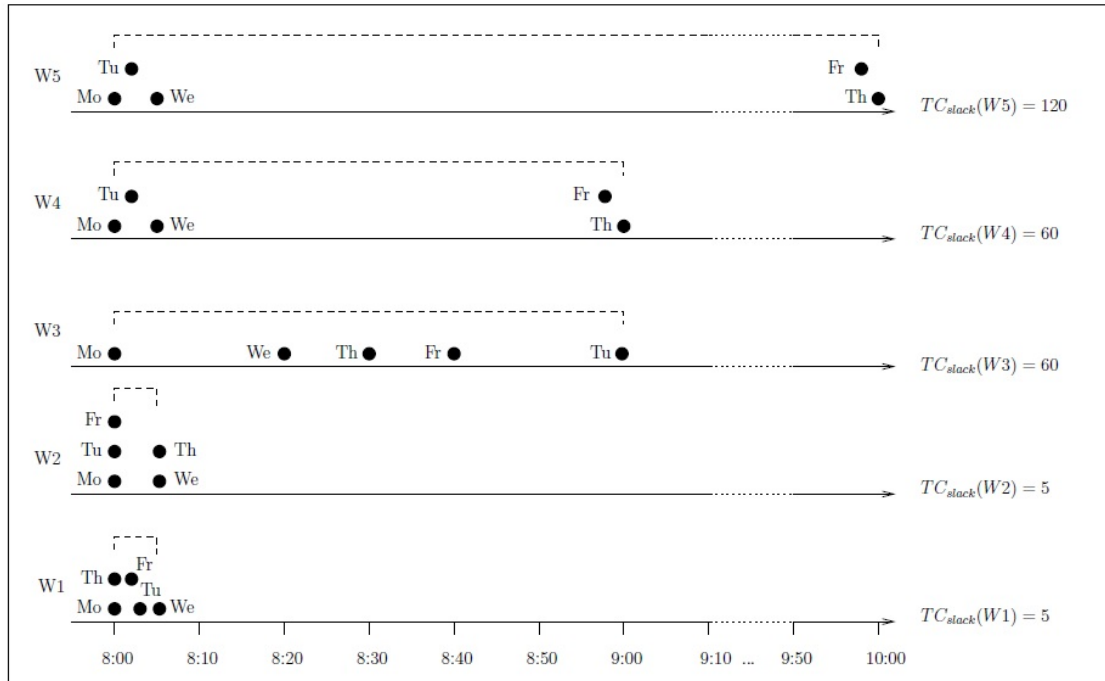


Figure 3.1 – Régularité des heures de passage et classes de temps

Soit $\delta > 0$ un paramètre qui exprime un écart entre deux horaires de service jugé comme négligeable par le client. On définit alors la notion de *classe horaire*.

Définition 3.1. Soit δ un réel strictement positif et D un horaire défini sur un horizon de plusieurs jours. Un sous-ensemble $C \subset D$ forme une classe horaire si pour toute paire $(d \in C, d' \in C)$, les horaires de passage w_d et $w_{d'}$ satisfont l'inégalité $|w_d - w_{d'}| \leq \delta$.

L'ensemble des heures de service d'un client sur l'horizon de temps peut donc facilement être décomposé en classe horaires. Cette décomposition n'est pas unique. Le nombre de classes horaires dans le planning d'un client modélise la régularité du service à ce client. Nous considérons la régularité d'une solution du problème de tournées comme le nombre maximal de classes horaires sur l'ensemble des clients.

3.3.2 Le Time-Consistent VRP

Nous décrivons maintenant le *Time-Consistent Vehicle Routing Problem* (TCVRP), qui inclut la notion de régularité pour chaque client. Soit $G = (V, A)$ un graphe complet où $V = \{0, \dots, n\}$ représente un dépôt (sommet 0) et un ensemble de n clients. On note $V^* = V \setminus \{0\}$. On considère une flotte homogène de K véhicules, de capacité Q , qui quittent le dépôt à l'heure 0 et transportent les clients jusqu'à leur destination. La demande des clients est caractérisée par les valeurs $p_i^d = 1$, valant 1 si le client i souhaite être transporté le jour d , et 0 sinon.

Le temps de transport $t_{ij} > 0$ est supposé connu pour chaque arc $(i, j) \in A$. Enfin, le paramètre δ représente l'écart maximal tel que deux heures de service soient perçues comme identiques.

On définit les variables de décision suivantes : x_{ijk}^d est une variable binaire valant 1 si l'arc (i, j) est utilisé par le véhicule k le jour d . w_i^d est l'heure de service au sommet i le jour d . w_{ik}^d est l'heure de service au sommet i le jour d s'il est transporté par le véhicule k . c_i^d est une variable entière représentant le numéro de la classe horaire du client i le jour d .

Des variables binaires intermédiaires $y_i^{d_1 d_2}$ sont fixées à 1 si le client i est servi plus tard le jour d_1 que le jour d_2 et que les deux horaires ne peuvent appartenir à la même classe, autrement dit $w_i^{d_2} - w_i^{d_1} > \delta$.

Notons que pour se focaliser sur la régularité, nous avons considéré une flotte homogène, nous n'avons pas considéré de durée de service ni de fenêtres horaires sur les clients. Nous avons ignoré les contraintes individuelles liées aux durées maximales de transport, aux fauteuils roulants et aux incompatibilités entre personnes. De même, un véhicule n'est pas autorisé à attendre. Dans le cas contraire, certains temps d'attente pourraient être introduits dans le seul but d'augmenter la régularité.

Le TCVRP consiste à trouver un ensemble d'au plus $|K|$ routes pour chaque jour de D , en minimisant les temps de transport et en respectant des contraintes de régularité sur les clients. Le modèle (3.1)-(3.19) présente une telle modélisation.

$$\text{Min } z = \sum_{d \in D} \sum_{k \in K} \sum_{(i,j) \in A} t_{ij} x_{ijk}^d \quad (3.1)$$

s.c.

$$\sum_{k \in K} \sum_{j \in V} x_{ijk}^d = p_i^d \quad (i \in V^*, d \in D), \quad (3.2)$$

$$\sum_{j \in V} x_{jik}^d - \sum_{j \in V} x_{ijk}^d = 0 \quad (i \in V^*, k \in K, d \in D), \quad (3.3)$$

$$\sum_{i \in V^*} x_{0ik}^d = 1 \quad (k \in K, d \in D), \quad (3.4)$$

$$\sum_{i \in V^*} x_{i0k}^d = 1 \quad (k \in K, d \in D), \quad (3.5)$$

$$\sum_{i \in V^*} \sum_{j \in V} x_{ijk}^d \leq Q \quad (k \in K, d \in D) \quad (3.6)$$

$$w_{jk}^d \leq w_{ik}^d + t_{ij} + (1 - x_{ijk}^d)M \quad ((i, j) \in A, k \in K, d \in D), \quad (3.7)$$

$$w_{jk}^d \geq w_{ik}^d + t_{ij} - (1 - x_{ijk}^d)M \quad ((i, j) \in A, k \in K, d \in D), \quad (3.8)$$

$$w_i^d \leq w_{ik}^d + (1 - \sum_{j \in V} x_{ijk}^d)M \quad (i \in V^*, k \in K, d \in D), \quad (3.9)$$

$$w_i^d \geq w_{ik}^d - (1 - \sum_{j \in V} x_{ijk}^d)M \quad (i \in V^*, k \in K, d \in D), \quad (3.10)$$

$$w_{0k}^d = 0 \quad (k \in K, d \in D), \quad (3.11)$$

$$w_i^{d_2} - w_i^{d_1} \leq \delta + M y_i^{d_1 d_2} \quad (i \in V^*, d_1, d_2 \in D), \quad (3.12)$$

$$c_i^{d_1} + 1 \leq c_i^{d_2} + M(1 - y_i^{d_1 d_2}) \quad (i \in V^*, d_1, d_2 \in D), \quad (3.13)$$

$$y_i^{d_1 d_2} \in \{0, 1\} \quad (i \in V^*, d_1, d_2 \in D), \quad (3.14)$$

$$c_i^d \leq C \quad (i \in V, d \in D), \quad (3.15)$$

$$x_{ijk}^d \in \{0, 1\} \quad ((i, j) \in A, k \in K, d \in D), \quad (3.16)$$

$$w_{ik}^d \geq 0 \quad (i \in V^*, k \in K, d \in D), \quad (3.17)$$

$$w_i^d \geq 0 \quad (i \in V^*, d \in D), \quad (3.18)$$

$$c_i^d \in \{1, \dots, C\} \quad (i \in V, d \in D). \quad (3.19)$$

Les contraintes (3.2) précisent qu'un client i ne peut être transporté le jour d que s'il émet une demande de transport pour ce jour. Les contraintes (3.3) expriment la conservation des flots en chaque sommet du graphe sauf le dépôt. Les contraintes (3.4) et (3.5) autorisent les véhicules à quitter et revenir au dépôt au plus une fois par jour. Les contraintes (3.6) modélisent la capacité des véhicules. Les contraintes (3.7) et (3.8) permettent de linéariser l'expression logique suivante :

$$\forall (i, j) \in A, \forall k \in K, \forall d \in D, x_{ijk}^d = 1 \Rightarrow w_{jk}^d = w_{ik}^d + t_{ij}.$$

Les contraintes (3.9) et (3.10) définissent les heures de service aux clients, indépendamment du véhicule utilisé. Elle linéarisent l'expression suivante :

$$\forall i \in V^*, \forall k \in K, \forall d \in D, \sum_{j \in V} x_{ijk}^d = 1 \Rightarrow w_i^d = w_{ik}^d.$$

Les contraintes (3.11) fixent l'heure de départ du dépôt. Pour chaque client, les contraintes (3.12)–(3.14) précisent que deux horaires de service qui diffèrent de plus de δ sont affectés à deux classes distinctes. Elles linéarisent l'expression suivante :

$$\forall i \in V^*, \forall d_1, d_2 \in D, w_i^{d_2} - w_i^{d_1} > \delta \Rightarrow c_i^{d_1} + 1 \leq c_i^{d_2}.$$

Les contraintes (3.15) limitent le nombre maximal de classes horaires par client.

Notons que lorsque $C = 1$, les contraintes (3.12)–(3.14), (3.15) et (3.19) peuvent être simplifiées en écrivant

$$|w_i^{d_2} - w_i^{d_1}| \leq \delta \quad (i \in V^*, d_1, d_2 \in D).$$

3.3.3 Une recherche à voisinage large pour le TCVRP

La régularité complique sérieusement le problème de tournées de véhicules car il ajoute des contraintes liant les périodes entre elles. Les tests réalisés avec le solveur Cplex 12.0 montrent que les instances de 10 à 12 requêtes proposées par Groër *et al.* (2009) nécessitent déjà un temps de calcul élevé. Nous recourons donc à une métaheuristique de type Recherche à Voisinage Large (LNS : Large Neighborhood Search) pour résoudre le TCVRP.

Le principe de la recherche à voisinage large consiste à détruire puis reconstruire itérativement des portions de solutions. Les premières utilisations de cette approche ont été proposées pour un problème de *Job-Shop* par Applegate et Cook (1991) et par Shaw (1998) pour le VRPTW dans une approche de programmation par contraintes. Le même principe a été proposé indépendamment par Schrimpf *et al.* (2000) sous le nom *Ruin and Recreate*.

Le fonctionnement général du LNS, dans le cadre d'un problème de minimisation, est décrit par l'Algorithme 2.

À chaque itération, un pourcentage des éléments constituant la solution courante est détruit (ligne 6). Dans un problème de tournées de véhicules, ces éléments sont par exemple des clients à router. Les éléments détruits sont ensuite réinsérés pour former une nouvelle solution (ligne 7). Plusieurs procédures ou *opérateurs* de destruction ou de reconstruction de la solution courante peuvent exister. Lorsque c'est le cas, le choix des opérateurs de destruction et de réparation à utiliser à l'itération courante se fait par une *roulette biaisée* (ligne 4).

L'algorithme 3 décrit notre mise en oeuvre du LNS pour résoudre le TCVRP. La solution est initialisée en résolvant un VRP pour chaque journée, avec une méthode de branch-and-price. La boucle principale (lignes 3 à 16) n'utilise qu'un opérateur de destruction, qui consiste tout simplement à supprimer toutes les routes d'une journée. L'opérateur de destruction sélectionne aléatoirement la journée à détruire (ligne 4), mais la journée sélectionnée n'est plus sélectionnable de nouveau avant que les autres autres journées n'aient été détruites à leur tour (ligne 14).

À la ligne 5, on choisit un client parmi ceux ayant un nombre maximal de classes. L'opérateur de reconstruction calcule de nouvelles routes pour cette journée, en cherchant à diminuer le nombre

Algorithme 2 Large Neighborhood Search (LNS)

Entrées: *SolutionInitiale*

```

1: MeilleureSolution  $\leftarrow$  SolutionInitiale
2: SolutionCourante  $\leftarrow$  SolutionInitiale
3: tant que le critère d'arrêt n'est pas respecté faire
4:   Sélection des heuristiques Destruction et Reconstruction
5:   S  $\leftarrow$  SolutionCourante
6:   S  $\leftarrow$  Destruction(S)
7:   S  $\leftarrow$  Reconstruction(S)
8:   si S meilleur que MeilleureSolution alors
9:     MeilleureSolution  $\leftarrow$  S
10:    SolutionCourante  $\leftarrow$  S
11:   sinon
12:     si S est accepté comme solution courante alors
13:       SolutionCourante  $\leftarrow$  S
14:     finsi
15:   finsi
16: fin tant que
17: retourne MeilleureSolution

```

de classes du client sélectionné et à contrôler le niveau de régularité de la nouvelle solution. Cela donne lieu à un sous-problème inédit, que nous appelons VRPmTW-nw (Vehicle Routing Problem with multiple Time Windows and no-wait constraints). Ce problème et sa résolution sont décrits dans la section 3.3.4.

Algorithme 3 Schéma général de l'algorithme LNS

```

1: sol  $\leftarrow$  solution initiale réalisable
2: D'  $\leftarrow$  D
3: répéter
4:   choisir un jour d au hasard dans D'
5:   Sélectionner un client i ayant le nombre de classes maximal sur le jour d
6:   Définir une instance du VRPmTW-nw pour le jour d en fonction de sol et i
7:   s  $\leftarrow$  solution du VRPmTW-nw ou null si solution non réalisable
8:   si s  $\neq$  null alors
9:     sol  $\leftarrow$  remplacer les routes du jour d par les routes de s
10:  finsi
11:  si D'  $\setminus$  {d}  $\neq$   $\emptyset$  alors
12:    D'  $\leftarrow$  D'  $\setminus$  {d}
13:  sinon
14:    D'  $\leftarrow$  D
15:  finsi
16: jusqu'à temps maximal atteint

```

3.3.4 Un sous-problème inédit : le VRPTW à fenêtres multiples et sans temps d'attente (VRPmTW-nw)

Considérons l'exemple d'un client dont les heures de service sont données par le vecteur $W = \{17, 34, 19, 45, 48\}$. Avec un paramètre de régularité δ fixé à 5, ce client possède trois classes horaires $\{17, 19\}$, $\{34\}$ et $\{45, 48\}$. Si l'opérateur de destruction détruit le mardi (34), le client ne possède plus que 2 classes horaires sur les jours restants. Supposons qu'on recherche une solution où le nombre de classes horaires pour chaque client soit au plus 2.

Une condition nécessaire et suffisante pour que le nombre de classes horaires après reconstruc-

tion reste à 2 est que le client soit servi dans les intervalles $[14, 22]$ ou $[43, 50]$. Le problème de VRPmTW-nw génère donc une double fenêtre horaire pour ce client. Considérons maintenant un client ne comportant qu'une classe horaire. Après destruction et reconstruction des routes du mardi, le nombre de classes horaires sera de nouveau 1 si la nouvelle heure de service est compatible avec les autres, ou 2 sinon. Dans les deux cas, ce client respecte le nombre maximal de classes horaires autorisé. Le VRPmTW-nw ne fixe donc aucune condition pour ce client.

Plus généralement, le VRPmTW-nw associe à chaque client une ou plusieurs fenêtres horaires en fonction du nombre de classes horaires ciblées. Le mécanisme utilisé pour fixer le nombre de classes horaires autorisées pour chaque client est décrit dans Feillet *et al.* (2014).

Le VRPmTW-nw peut être résolu grâce à une adaptation d'une méthode de type branch-and-price conçue à l'origine pour le VRPTW (Desaulniers *et al.* (2005); Feillet (2010)). Cette méthode permet en théorie de résoudre le VRPmTW-nw de manière optimale. Comme ce problème doit être résolu à chaque itération du LNS, il est capital de ne pas y consacrer trop de temps de calcul. Nous avons donc opté pour une version heuristique du branch-and-price en réduisant le nombre de colonnes générées, en résolvant de manière sous-optimale le problème de pricing et en explorant partiellement l'arbre des solutions. Le détail de l'heuristique et le paramétrage utilisé sont décrits dans Feillet *et al.* (2014).

3.3.5 Tests numériques

Nous avons tout d'abord résolu le TCVRP sur un ensemble de 10 instances proposées par Groër *et al.* (2009) avec 10 ou 12 clients et 3 journées. Sur ces instances, l'algorithme LNS est toujours capable de trouver des solutions à 1, 2 ou 3 classes horaires, alors que Cplex ne trouve des solutions à 1 classe horaire que sur 2 des 10 instances. Lorsque Cplex trouve une solution optimale, le LNS la trouve quasiment toujours en moins de 5 minutes de calcul. Lorsque Cplex ne trouve pas de solution optimale, le LNS trouve de meilleures solution que Cplex dans trois cas.

Nous avons ensuite généré 70 instances sur la base de 14 fichiers d'adresses, de 15 à 65 personnes, fournies par l'ADAPEI44. Pour chacune des 14 jeux de données, nous avons généré aléatoirement 5 profils de demande sur la semaine, avec des taux moyens de présence variant entre 50% et 90%. L'algorithme a été exécuté pendant une heure de calcul, ce qui représente une moyenne de 6000 itérations, ce que nous avons jugé suffisant pour explorer convenablement l'espace des solutions (à chaque itération, jusqu'à 20% d'une solution peut être modifiée).

Le Tableau 3.1 résume les résultats obtenus pour la résolution du TCVRP. Chaque instance a été résolue en donnant successivement les valeurs 5, 4, 3, 2 et 1 à la valeur C dans les contraintes 3.15. Le coût moyen de la régularité est la différence relative entre les fonctions objectif pour $C = 5$ et toute valeur C inférieure à 5.

	Nombre maximal de classes horaires				
	5	4	3	2	1
Solutions obtenues	70/70	70/70	70/70	70/70	70/70
Fonction objectif (moyenne)	1561.93	1561.95	1563.50	1568.14	1660.65
Coût moyen de la régularité	-	0.00%	0.10%	0.30%	5.90%

Tableau 3.1 – Résultats expérimentaux sur 70 instances

L'heuristique permet de trouver des solutions réalisables pour chacune des 70 instances. Les instances les plus difficiles à résoudre sont celles comportant 70% ou 80% de taux de présence des clients. En effet, lorsque seulement 50% des clients sont transportés, il est facile de modifier les tournées de façon à proposer des horaires réguliers. Lorsque les clients ont un taux de présence moyen de 90%, cela signifie que de nombreux clients sont présents tous les jours, et donc les tournées

sont naturellement régulières. En revanche, il est plus difficile de traiter les cas intermédiaires. Le besoin de régularité ne semble pas pénaliser énormément le coût jusqu'à 2 classes horaires. Si on impose une contrainte stricte d'une classe horaire par client, le surcoût est de 5.90% en moyenne.

Une première extension possible de ces travaux consiste à résoudre de manière exacte le problème à 1 classe horaire, en adaptant la méthode de branch-and-price de Feillet (2010). Une seconde extension serait de générer des tournées irrégulières, pour des applications telles que le transport de fonds.

3.4 Projet MCDARP : Problème de type Dial-A-Ride multi-critères

Le projet Marika optimise les tournées de véhicules établissement par établissement. Or, il existe souvent plusieurs établissements très proches les uns des autres et accueillant un public d'une zone géographique quasiment identique. La mutualisation des transports entre ces établissements permettrait alors d'économiser une part non négligeable des frais de transport. Cela nécessite de modéliser le problème de transport comme un Pickup ou Delivery Problem (PDP) ou un Dial-a-Ride Problem (DARP) au lieu d'un VRP.

De plus, selon les parties prenantes (établissements, financeurs, passagers, transporteurs) un plan de transport peut être évalué de différentes manières. La conception des tournées est donc abordée comme un problème multi-critères. Cette section présente les travaux réalisés dans le cadre du projet projet MCDARP (Multi Criteria Dial-A-Ride Problem), projet Hubert Curien en partenariat avec l'Université de Vienne (Autriche) en 2009 et 2010. Ces travaux ont été publiés dans le Journal of the Operational Research Society (Lehuédé *et al.*, 2014).

A notre connaissance, les travaux sur le DARP multi-critères se sont majoritairement concentrés sur la recherche d'un ensemble de solutions non dominées constituant une bonne approximation du front de Pareto (voir par exemple Paquette *et al.* (2013) ou Atahran *et al.* (2014)). En présence d'un nombre potentiellement élevé d'objectifs, nous privilégions une approche multi-critères qui modélise les préférences de l'utilisateur et agrège les différents critères sous la forme d'une intégrale de Choquet (Grabisch et Labreuche (2010)). Cette approche est particulièrement pertinente lorsqu'il existe de fortes interactions entre les objectifs. Il en résulte un programme linéaire mixte avec une seule fonction objectif, résolu par une recherche à voisinage large (LNS).

Dans cette section, nous décrivons tout d'abord le problème de type DARP à résoudre, puis nous présentons le modèle multi-critères et la formulation complète du problème. Nous présentons ensuite très rapidement les opérateurs utilisés dans la recherche à voisinage large et quelques résultats expérimentaux.

3.4.1 Description du problème mutualisé de transport

Soit R un ensemble de requêtes à satisfaire. Chaque requête $i \in R$ est déterminée par un point de collecte (pickup) noté $p(i)$, un point de livraison (delivery) noté $d(i)$ et une quantité q_i de personnes à transporter entre ces lieux. On note P l'ensemble des points de collecte et D l'ensemble des points de livraison. On suppose une flotte homogène de K véhicules de capacité \bar{Q} . On note O et O' les ensembles des dépôts initiaux et terminaux des véhicules. Le véhicule k commence son itinéraire du dépôt $o(k) \in O$ et le termine en $o'(k) \in O'$. On définit $N = P \cup D$ l'ensemble des sites de collecte ou livraison.

Le DARP multi-critères est défini sur un graphe $G = (V, A)$ où $V = P \cup D \cup O \cup O'$ est l'ensemble de tous les sites visités par les véhicules et A est l'ensemble de tous les arcs (i, j) avec $i \in V \setminus O'$ et $j \in V \setminus O$. On note t_{ij} le temps de transport sur l'arc $(i, j) \in A$. Le DARP implique généralement des limitations sur les temps de transport. On considère ici une limite \bar{T} sur le temps de trajet de chaque véhicule, et une limite \bar{L}_i sur le temps de transport de la requête $i \in R$.

En chaque sommet $i \in N$ est définie une durée de service s_i et une fenêtre horaire $[e_i, l_i]$. Si un véhicule arrive à une destination avant la fenêtre horaire, il doit en attendre l'ouverture. En revanche, on n'autorise pas l'attente d'un véhicule à l'intérieur d'une fenêtre horaire. Une fois descendus d'un véhicule, les passagers en avance doivent attendre l'heure d'ouverture l_i de leur établissement.

Le DARP multi-critères (MCDARP) consiste à déterminer un ensemble de routes permettant de servir la totalité des requêtes en optimisant plusieurs critères liés à l'efficacité du système de transport et à la qualité de service aux passagers.

Ce problème s'appuie sur les variables binaires x_{ij}^k , valant 1 si l'arc (i, j) est utilisé par le véhicule k , et 0 sinon. Pour clarifier la présentation du modèle, nous introduisons également les variables y_i^k , valant 1 si la requête i est servie par le véhicule k et 0 sinon. Le modèle comporte 4 ensembles de variables continues : A_i^k est l'heure d'arrivée du véhicule k au sommet $i \in V$, B_i^k est l'heure de début de service du sommet $i \in N$ par le véhicule k , L_i est le temps de trajet de la requête $i \in R$, Q_i^k est le nombre de passagers du véhicule k lorsqu'il quitte le sommet $i \in N$. En pratique, comme les demandes sont entières, les variables Q_i^k prennent des valeurs entières.

Le modèle multi-critères

L'aide à la décision multi-critères traite des problèmes d'optimisation sur la base d'une modélisation des préférences des décideurs par rapport à un certain nombre de critères. Considérant un nombre \aleph de critères, le décideur exprime son degré de satisfaction en fonction d'un ensemble de métriques $\Omega_i, i \in \{1, \dots, \aleph\}$, qui jouent le même rôle qu'une fonction objectif. On introduit des fonctions d'utilité $u_i : \Omega_i \rightarrow [0, 1]$, qui expriment la qualité z_i d'une solution par rapport à un critère i . Plus l'utilité est proche de 1, plus la solution est satisfaisante pour le critère i .

Une solution x est donc caractérisée par un vecteur $(z_1, \dots, z_{\aleph}) \in \Omega_1 \times \dots \times \Omega_{\aleph}$ et évaluée par une fonction d'agrégation $H = H(u_1(z_1), \dots, u_{\aleph}(z_{\aleph})) : [0, 1]^{\aleph} \rightarrow [0, 1]$ (Keeney et Raiffa (1976)). Nous proposons de calculer H comme une intégrale de Choquet, qui a pour intérêt de considérer non seulement l'importance relative des critères, mais aussi les possibles interactions positives ou négatives entre critères. L'intégrale de Choquet pour un vecteur $(z_1, \dots, z_{\aleph}) \in [0, 1]^{\aleph}$ est définie par Grabisch (1997) comme :

$$H = \sum_{i=1, \dots, \aleph} \left(\phi_i - \frac{1}{2} \sum_{j \neq i} |I_{ij}| \right) z_i + \sum_{I_{ij} > 0} I_{ij} \min(z_i, z_j) + \sum_{I_{ij} < 0} |I_{ij}| \max(z_i, z_j) \quad (3.20)$$

où ϕ_i est l'importance relative du critère i et I_{ij} mesure l'interaction entre les critères i et j . $H(z_1, \dots, z_{\aleph})$ comprend une partie linéaire, un terme *conjunctif* et un terme *disjonctif*. Dans la partie conjunctive, une interaction positive entre deux termes dénote la complémentarité entre ces critères. L'opérateur min exprime le fait que les deux critères complémentaires doivent tous deux avoir un score élevé pour que le terme soit élevé. En revanche, dans le terme disjonctif, deux critères ayant une interaction négative produisent un score élevé dès qu'un critère est bien satisfait ; on parle alors de *substituabilité* entre critères. Il est à noter que les poids de z_i , $\min(z_i, z_j)$ et $\max(z_i, z_j)$ dans (3.20) sont non-négatifs et ont pour somme 1. Une description plus complète de ce modèle de préférence peut être trouvée dans l'état de l'art de Grabisch et Labreuche (2010).

Métriques pour le MCDARP

Plus que d'autres problèmes d'optimisation des transports, le DARP se prête à une modélisation multi-critères. Parmi les objectifs les plus classiques relevés dans la littérature, on peut citer ceux liés au coût (nombre de véhicules utilisés, distance parcourue, etc.) et ceux liés à la qualité de service (temps de trajet, temps d'attente etc.). Sur ce sujet, nous renvoyons le lecteur à la revue de la littérature de Paquette *et al.* (2009).

Au cours du projet Marika, nous avons identifié 5 objectifs pertinents. Soit X l'ensemble des solutions du MCDARP et $x \in X$ une solution particulière. On définit :

$$\begin{aligned}
f_{mrt}(x) &= \max_{i \in R} (L_i - t_{p(i),d(i)}) && (\text{détour maximal}), \\
f_{art}(x) &= \frac{1}{\sum_{i \in R} q_i} \sum_{i \in R} q_i (L_i - t_{p(i),d(i)}) && (\text{détour moyen}), \\
f_{wtd}(x) &= \sum_{i \in R} q_i (l_{d(i)} - B_{d(i)}^k) && (\text{temps d'attente total à destination}), \\
f_{tt}(x) &= \sum_{k \in K} \sum_{ij \in V} t_{ij} x_{ij}^k && (\text{durée d'utilisation des véhicules}), \\
f_v(x) &= \sum_{k \in K} \sum_{i \in P} x_{o(k),i}^k && (\text{nombre de véhicules}).
\end{aligned}$$

Les fonctions f_{mrt} et f_{art} sont basées sur la notion de détour. Il est à noter que les $t_{p(i),d(i)}$ sont des valeurs constantes, et donc minimiser le détour moyen ou la somme des temps de trajet est équivalent. La fonction f_{wtd} mesure le temps d'attente des passagers entre leur date $B_{d(i)}^k$ d'arrivée à destination et la fin $l_{d(i)}$ de la fenêtre horaire, qui correspond à l'ouverture de l'établissement. Les fonctions f_{tt} et f_v sont utilisées comme indicateurs du coût de la solution, qui inclut souvent une part fixe liée à l'utilisation des véhicules et une part variable liée à leur durée d'utilisation.

Contrairement à une fonction objectif classique, une métrique dans un modèle multi-critères se doit d'être indépendante de la taille des instances. On procède donc à une normalisation des trois derniers objectifs. Les métriques utilisées dans nos travaux sont définies par :

$$\begin{aligned}
(\Omega_1) \quad m_{mrt}(x) &= f_{mrt}(x) && (\text{détour maximal}), \\
(\Omega_2) \quad m_{art}(x) &= f_{art}(x) && (\text{détour moyen}), \\
(\Omega_3) \quad m_{wtd}(x) &= \frac{f_{wtd}(x)}{Q} && (\text{temps d'attente moyen à destination}), \\
(\Omega_4) \quad m_{tt}(x) &= \frac{f_{tt}(x)}{Q} && (\text{durée moyenne d'utilisation d'un véhicule}), \\
(\Omega_5) \quad m_v(x) &= \frac{Q}{f_v(x)Q} && (\text{taux d'occupation moyen d'un véhicule}).
\end{aligned}$$

Q représente le nombre total de passagers à transporter. $m_v(x)$ peut être supérieur à 1 car un siège peut être occupé par plusieurs personnes successivement.

Les fonctions d'utilité pour chacun des critères sont construites comme des fonctions linéaires par morceaux, représentés sur la Figure 3.2. Ces fonctions fournissent des valeurs comprises entre 0 et 1 pour chaque valeur possible de la métrique associée.

Fonctions objectif basées sur les préférences des décideurs

Selon la valeur de l'importance relative ϕ_i des critères et des interactions I_{ij} entre critères, il est possible de modéliser les préférences de plusieurs types de décideurs. Les Tableaux 3.2 et 3.3 donnent deux exemples de modélisations.

Le premier modèle, noté μ_{cost} , reflète les préférences du prestataire de transport, qui se focalise principalement sur le coût. Cela se traduit par des poids élevés sur les deux derniers critères (0.4 et 0.45 respectivement). Les critères liés à la qualité de service ont une importance individuelle faible (0.09, 0.04, et 0.02). De plus, toutes les interactions positives impliquent au moins un critère lié au coût. Autrement dit, si une solution est mauvaise du point de vue des coûts, augmenter la qualité de service ne permettra d'augmenter la fonction objectif que de manière très limitée.

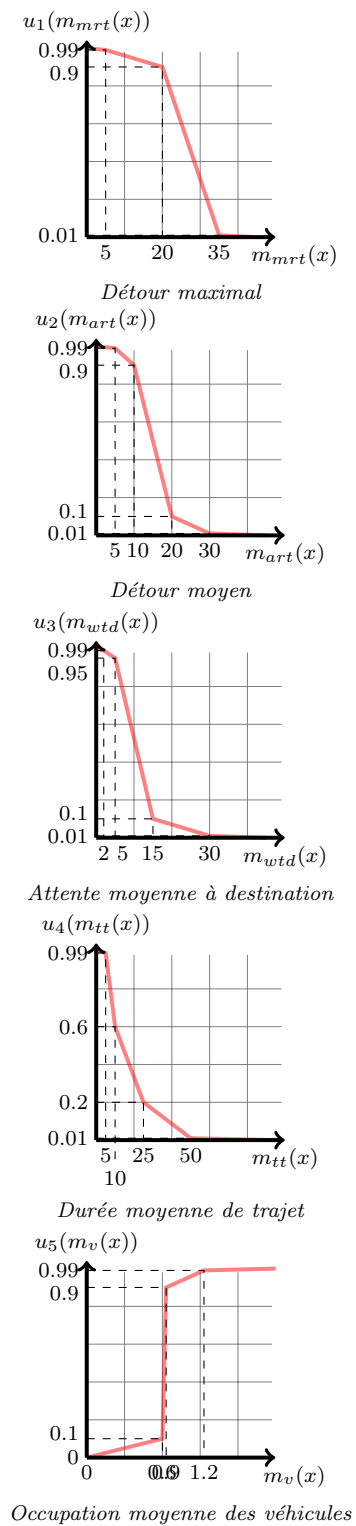


Figure 3.2 – Fonctions d'utilité partielles pour les critères du MCDARP

	ϕ_{mrt}	ϕ_{art}	ϕ_{wtd}	ϕ_{tt}	ϕ_v
μ_{cost}	0.09	0.04	0.02	0.4	0.45
μ_{QoS}	0.35	0.3	0.2	0.075	0.075

Tableau 3.2 – Importance des critères

	$I_{mrt,art}$	$I_{mrt,wtd}$	$I_{mrt,tt}$	$I_{mrt,v}$	$I_{art,wtd}$	$I_{art,tt}$	$I_{art,v}$	$I_{wtd,tt}$	$I_{wtd,v}$	$I_{tt,v}$
μ_{cost}	0	0	0.08	0.1	0	0.02	0.05	0	0.04	0.5
μ_{QoS}	0.3	0.2	0.1	0.1	0.2	0	0	0	0	0

Tableau 3.3 – Interaction entre critères

Le deuxième modèle, noté μ_{QoS} , reflète les préférences d'un décideur sensible à la qualité de service aux passagers. Ainsi, les trois premiers critères ont un poids respectif de 0.35, 0.3 et 0.2. Les critères liés au coût ont un poids de 0.075 chacun. Un poids de 0.3 est donné pour l'interaction entre les deux premiers critères. En revanche, de nombreuses interactions entre critères ont un poids nul, et n'ont donc pas d'impact sur la fonction objectif.

3.4.2 Un modèle mathématique pour le MCDARP

Nous pouvons maintenant introduire le modèle mathématique du MCDARP :

$$\max z = H(u_1(m_{mrt}(x)), u_2(m_{art}(x)), u_3(m_{wtd}(x)), u_4(m_{tt}(x)), u_5(m_v(x))) \quad (3.21)$$

S.C.

$$\sum_{k \in K} y_i^k = 1 \quad \forall i \in R, \quad (3.22)$$

$$\sum_{j \in N} x_{p(i),j}^k = y_i^k \quad \forall i \in R, k \in K, \quad (3.23)$$

$$\sum_{j \in N} x_{p(i),j}^k - \sum_{j \in N} x_{j,d(i)}^k = 0 \quad \forall i \in R, k \in K, \quad (3.24)$$

$$\sum_{j \in P \cup \{o'(k)\}} x_{o(k),j}^k = 1 \quad \forall k \in K, \quad (3.25)$$

$$\sum_{i \in D \cup \{o(k)\}} x_{i,o'(k)}^k = 1 \quad \forall k \in K, \quad (3.26)$$

$$\sum_{i \in V} x_{ij}^k - \sum_{i \in V} x_{ji}^k = 0 \quad \forall j \in N, k \in K, \quad (3.27)$$

$$x_{ij}^k = 1 \Rightarrow Q_j^k \geq Q_i^k + q_j \quad \forall i \in V, j \in P, k \in K \quad (3.28)$$

$$x_{ij}^k = 1 \Rightarrow Q_j^k \geq Q_i^k - \sum_{i' \in R, d(i')=j} y_{i'}^k q_{i'} \quad \forall i \in N, j \in D, k \in K, \quad (3.29)$$

$$0 \leq Q_i^k \leq \bar{Q} \quad \forall i \in N, k \in K, \quad (3.30)$$

$$x_{ij}^k = 1 \Rightarrow A_j^k = B_i^k + s_i + t_{ij} \quad \forall i, j \in V, k \in K, \quad (3.31)$$

$$B_i^k \geq A_i^k \quad \forall i \in V, k \in K, \quad (3.32)$$

$$\sum_{i \in N} Q_i^k x_{ij}^k \geq 1 \wedge A_j^k \geq e_j \Rightarrow B_j^k = A_j^k \quad \forall j \in N, k \in K, \quad (3.33)$$

$$y_i^k (B_{d(i)}^k - B_{p(i)}^k - s_{p(i)}) \leq L_i \quad \forall i \in R, k \in K, \quad (3.34)$$

$$t_{p(i),d(i)} \leq L_i \leq \bar{L}_i \quad \forall i \in R, \quad (3.35)$$

$$B_{o'(k)}^k - B_{o(k)}^k \leq \bar{T} \quad \forall k \in K, \quad (3.36)$$

$$e_i \leq B_i^k \leq l_i \quad \forall i \in V, k \in K, \quad (3.37)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in V, k \in K, \quad (3.38)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in R, k \in K, \quad (3.39)$$

$$Q_i^k \geq 0 \quad \forall i \in N, k \in K. \quad (3.40)$$

La fonction objectif (3.21) minimise l'intégrale de Choquet associée aux cinq critères introduits plus haut. Les contraintes (3.22) garantissent que chaque requête soit servie par un véhicule unique. Les contraintes (3.23) et (3.24) assurent que les point de montée et de descente d'une requête soient visitées par le même véhicule. Les contraintes (3.25) et (3.26) vérifient que chaque véhicule quitte son dépôt initial et revienne à son dépôt final une seule fois. Lorsqu'un véhicule k n'est pas utilisé, un trajet à coût nul de $o(k)$ à $o'(k)$ permet de satisfaire les contraintes (3.25) et (3.26). Les contraintes (3.27) garantissent que les véhicules arrivant en un point de montée ou de descente puissent en repartir. Les contraintes (3.28) permettent le calcul des charges des véhicules. Les contraintes (3.29) assurent qu'un nombre correct de personnes quittent un véhicule au point de descente. Les contraintes (3.30) vérifient que la capacité des véhicules est respectée. Les contraintes (3.31) fixent les heures d'arrivée et les contraintes (3.32) assurent qu'un service en un point ne commence pas avant l'arrivée d'un véhicule. Les contraintes (3.33) empêchent l'attente d'un véhicule en un point de montée lorsqu'il a déjà des passagers à son bord. Les contraintes (3.34) et (3.35) calculent les temps de trajet des passagers et vérifient si ces temps de trajet sont acceptables. Les contraintes (3.36) vérifient si la durée maximale des routes n'est pas dépassée. Les contraintes (3.37) garantissent le respect des fenêtres horaires. Les contraintes (3.28), (3.29), (3.31), (3.33), (3.34) peuvent être linéarisées (voir Parragh (2009)).

3.4.3 Méthode de résolution

Nous avons résolu le modèle MCDARP avec une méthode adaptative à voisinage large (Adaptive Large Neighborhood Search (ALNS)) inspirée de Ropke et Pisinger (2006). Par rapport à l'algorithme 2, la méthode utilisée pour le MCDARP utilise de nombreux opérateurs de destruction et de reconstruction. La méthode adaptative rajoute donc une évaluation de chaque opérateur de destruction ou de reconstruction, basée sur ses performances aux itérations précédentes. Ainsi, chaque opérateur reçoit un score qui est régulièrement mis à jour. La probabilité de choisir un opérateur à l'itération courante dépend de ce score. Nous utilisons 3 catégories d'opérateurs : des opérateurs déjà existants dans la littérature, des opérateurs spécifiques au DARP multi-critères et les opérateurs dédiés à un des cinq critères d'optimisation.

Opérateurs issus de la littérature

1. *Destruction aléatoire (Random removal)* : cet opérateur choisit des requêtes aléatoirement et les retire de la solution courante.
2. *Destruction "au pire" (Worst removal)* (Ropke et Pisinger, 2006) : on calcule la contribution de chaque requête à la fonction objectif en calculant la valeur d'une solution sans cette requête. Chaque requête reçoit ensuite une probabilité d'être sélectionnée qui croît avec sa contribution.
3. *Destruction selon la distance (Distance related removal)* : cet opérateur cherche à retirer un ensemble de requêtes similaires. Nous implémentons la version de Ropke et Pisinger (2006) : une requête initiale i est sélectionnée aléatoirement et retirée de la solution courante. Ensuite nous calculons une mesure de similarité r_{ij} entre i et chaque autre requête $j \neq i$:

$$r_{ij} = \frac{1}{4}(t_{p(i),p(j)} + t_{d(i),p(j)} + t_{p(i),d(j)} + t_{d(i),d(j)}).$$

Chaque requête j reçoit ensuite une probabilité d'être sélectionnée qui croît avec r_{ij} .

4. *Destruction selon les heures de service (Time related removal)* : cet opérateur cherche à supprimer un ensemble de requêtes pouvant être servies aux mêmes heures. Une heure t_s est tout d'abord choisie aléatoirement, puis les clients sont triés par ordre croissant de la valeur v_i définie par

$$v_i = \begin{cases} e'_i - t_s & \text{si } t_s \leq e'_i \\ 0 & \text{si } e'_i \leq t_s \leq l'_i \\ t_s - l'_i & \text{si } t_s > l'_i. \end{cases}$$

L'intervalle $[e'_i, l'_i]$ correspond aux heures de visite possibles de i préservant la réalisabilité de sa route (Prescott-Gagnon *et al.*, 2009).

5. *Destruction selon la distance et les heures de service (Proximity based removal)* (Prescott-Gagnon *et al.*, 2009) : cet opérateur fonctionne de manière similaire à celui basé sur les distances, mais inclut également des informations relatives aux heures de passage.
6. *Meilleure insertion* (Ropke et Pisinger, 2006) : cet opérateur consiste à insérer les requêtes détruites à leur meilleure position d'insertion possible parmi toutes les routes actives. Par rapport à un problème de DARP classique, plusieurs requêtes partagent ici la même point de destination. Il est donc fréquent que le point de destination d'une requête soit déjà présent sur une ou plusieurs routes.
7. *Insertion avec regret* : on appelle κ -regret la somme des différences entre la meilleure insertion possible et les deuxième, troisième... meilleures insertions. Si une requête n'est pas insérable, le regret est fixé à 0. Tout comme Ropke et Pisinger (2006), nous utilisons plusieurs opérateurs d'insertion avec regret (avec $\kappa \in \{2, 3, 4\}$).

Opérateurs spécifiques au MCDARP

1. *Retrait d'une destination* : cet opérateur consiste à retirer un point de destination d'une route. Dans une route donnée, on choisit le point de destination auquel le moins de requêtes sont associées. Le point de destination est retiré de la route, ainsi que tous les points de montée des requêtes associées.
2. *Retrait de cluster à partir d'un point de montée* : on choisit une route et un point de montée p au hasard. A partir de p , on définit comme cluster l'ensemble des points de montée suivant p de manière ininterrompue et dont les requêtes partagent la même destination que p . On retire ainsi des clusters jusqu'à avoir retiré suffisamment de requêtes.
3. *Retrait de route* : on retire des routes au hasard, jusqu'à avoir retiré un pourcentage satisfaisant de requêtes. Cet opérateur est utile pour déconstruire des routes incluant plusieurs destinations.

Opérateurs guidés par un critère

L'ensemble des opérateurs précédents reposent sur le calcul d'une intégrale de Choquet. Afin d'épargner du temps de calcul, nous avons proposé plusieurs opérateurs guidés par un seul des critères d'optimisation. Ainsi le critère à optimiser peut être considéré comme un *paramètre* de l'opérateur. On parle alors d'*opérateurs paramétrés*. Nous avons implémenté une version paramétrée du *worst removal*, de la *meilleure insertion* et de la *meilleure insertion avec regret*.

3.4.4 Expérimentations numériques

L'heuristique a été codée en C++ et exécutée sur un processeur 2.67 GHz avec 24GB de RAM. Trois variantes ont été testées.

- *Basique* : nous utilisons toutes les combinaisons entre les opérateurs de destruction {random, worst, time related, distance related, proximity} et les opérateurs de reconstruction {best insertion, 2-regret, 3-regret, 4-regret}.
- *Avancé* : cette variante utilise tous les opérateurs de la variante précédente, et ajoute des opérateurs spécifiques au MCDARP. Plus précisément, nous ajoutons toutes les combinaisons possibles entre les opérateurs de destruction {route delivery, clustered pickups} et les opérateur de réparation {best insertion, 2-regret, 3-regret, 4-regret}.
- *Paramétré* : dans cette variante, nous ajoutons les opérateurs paramétrés.

L'algorithme a été évalué sur 14 instances fournies par l'ADAPEI44 et le Conseil Général de Loire-Atlantique. Les instances ADAPEI44 sont construites par regroupement de plusieurs instances du projet Marika. La plus grande de ces instances comporte 166 requêtes et 4 établissements. Les instances fournies par le Conseil Général de Loire-Atlantique concernent le transport d'élèves handicapés vers des établissements scolaires du département. Ces instances se caractérisent donc par un nombre beaucoup plus élevé de destinations et un faible nombre de requêtes par destination. La plus grande instance comporte 200 requêtes pour 51 établissements.

L'article Lehuédé *et al.* (2014) décrit largement les tests numériques réalisés. Comme il est difficile d'interpréter et de comparer des valeurs d'intégrales de Choquet, nous concluons cette section par une représentation graphique d'une telle intégrale.

La Figure 3.3 représente l'intégrale de Choquet pour deux solutions d'une même instance. Chaque portion du graphique représente un terme de l'intégrale de Choquet. Les cinq critères sont notés par v (nombre de véhicules), tt (travel time), art (détour moyen), mrt (détour maximal) et wtd (attente à destination). Les interactions positives en deux critères $c1$ et $c2$ sont notées $(c1, c2)$. (Il n'y a pas d'interaction négative dans notre modèle). L'angle de chaque portion est proportionnel à la valeur du coefficient dans l'équation (3.20). Le pourcentage de hauteur colorée représente la

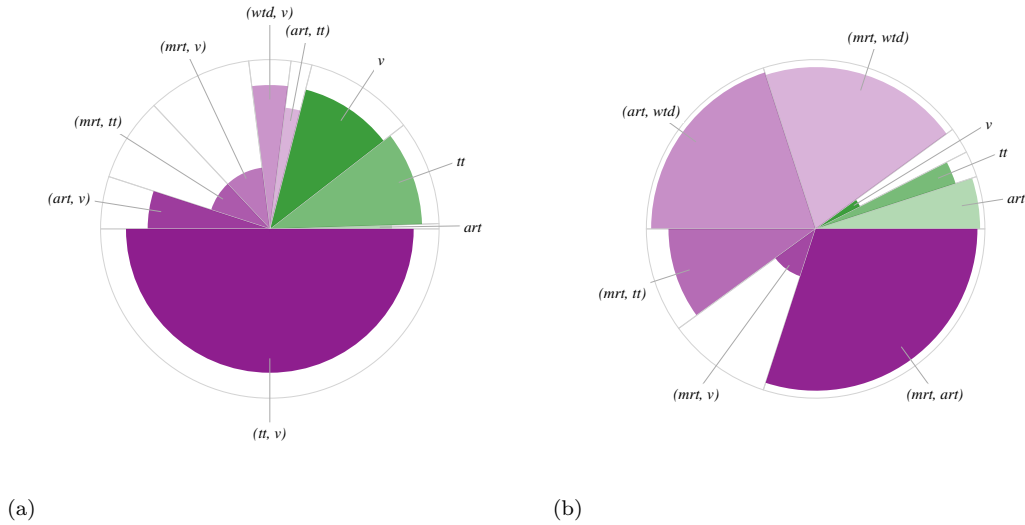


Figure 3.3 – Représentations de l'intégrale de Choquet

valeur de z_i ou $\min(z_i, z_j)$. Ainsi, la proportion colorée dans le cercle donne la valeur de l'intégrale de Choquet.

La partie gauche (a) de la figure représente une solution obtenue pour un décideur focalisé sur le coût. la partie droit (b) de la figure représente une solution obtenue pour un décideur focalisé sur la qualité de service.

3.5 Conclusions

Dans ce chapitre, nous avons présenté un problème d'optimisation du transport de personnes handicapées mentales, qui a donné lieu à un contrat de valorisation, ainsi que deux extensions de ce problème qui ont fait l'objet de projets de recherche.

Les modèles et algorithmes décrits dans ce chapitre sont intégrables séparément dans un même outil logiciel pour l'optimisation du transport de personnes. En revanche, les mettre en oeuvre simultanément nécessiterait de traiter à la fois l'aspect multi-périodique et multi-objectif. Une fois encore, il s'agit de trouver un bon compromis entre la qualité de la réponse à l'utilisateur et le temps de calcul nécessaire. De multiples contraintes métiers, potentiellement différentes d'un utilisateur à l'autre devraient être incorporées. Par exemple, les travaux décrits dans ce chapitre n'ont pas pris en compte les trajets aller et retour : les établissements s'efforcent souvent de pas infliger les détours les plus longs aux mêmes personnes le matin et l'après midi.

Un problème intéressant qui se pose à un niveau plus global est la sectorisation de la demande de transport. Les conseils généraux (chargés de l'organisation des transports scolaires) et les organismes régissant un nombre élevé d'établissements médico-sociaux lancent des appels d'offre découpés en lots regroupant des sous-ensembles d'établissements. Combiner les problèmes de partition de territoire et de mutualisation des transports serait une extension pertinente à la fois du point de vue théorique et pratique.

Chapitre 4

Problèmes de transport avec transferts de passagers

4.1 Introduction

Une mutualisation complète des transports entre plusieurs établissements peut conduire à transférer certains passagers d'un véhicule à un autre. Ces passagers sont alors pris en charge par un véhicule à leur domicile et conduits jusqu'à un point de transfert (typiquement un autre établissement). Là, il embarquent dans un deuxième véhicule qui les conduit à destination. Cette pratique occasionne une rupture de charge mais permet de rationaliser l'usage des véhicules.

Cette classe de problèmes, appelée *problèmes de collectes et livraisons avec transferts* (PDPT), a fait l'objet de la thèse de doctorat de Renaud Masson (2009–2012), co-encadrée avec Fabien Lehuédé, et dirigée par Nathalie Bostel.

Par rapport au PDP classique, le PDPT relaxe la contrainte imposant qu'un même véhicule prenne en charge les passagers à leur point de départ et d'arrivée. On considère donc un ensemble de requêtes, chacune étant identifiée par un point de collecte, un point de livraison et un nombre de personnes à transporter. Les points de collecte et de livraison doivent être desservis dans des fenêtres de temps prédéfinies. Les requêtes peuvent être transférées d'un véhicule à un autre à des emplacements spécifiques, appelés *points de transfert*. Dans le cas du transport de personnes handicapées mentales, un encadrement des personnes doit être présent aux points de transfert. Ces derniers sont donc positionnés sur les points de livraison.

L'objectif du PDPT est de construire, sur une seule période, un ensemble de tournées pour satisfaire les requêtes à coût minimal. Pour des raisons de qualité de service, chaque requête peut être transférée au plus une fois. L'exemple illustré par les Figures 4.1 et 4.2 montre l'intérêt de recourir aux transferts.

La Figure 4.1 représente la solution optimale d'un problème de type PDP n'utilisant pas le point de transfert τ . Dans la Figure 4.2, les requêtes 3 et 6 sont échangées au point de transfert τ , ce qui permet d'économiser environ 20% de la distance totale parcourue. Le temps de trajet des requêtes 3, 4, 5 et 6 est raccourci, de même que la distance totale. En contrepartie, le temps de trajet des requêtes 1 et 2 est allongé et les requêtes 3 et 6 doivent changer de véhicule au point de transfert. De plus, une telle solution impose une synchronisation des véhicules, qui se donnent rendez-vous au point de transfert. Les deux tournées, auparavant indépendantes sont maintenant liées. Cette interconnexion des tournées a des conséquences à la fois sur la modélisation du problème et sur les algorithmes de résolution. Ces conséquences seront détaillées au long de ce chapitre.

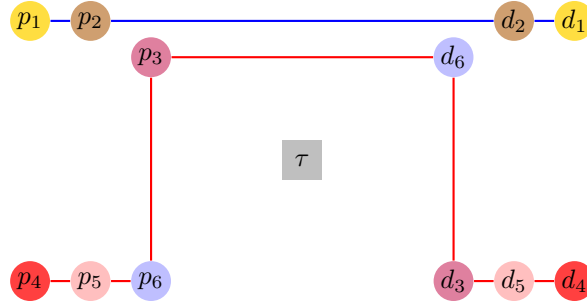


Figure 4.1 – Solution sans transfert

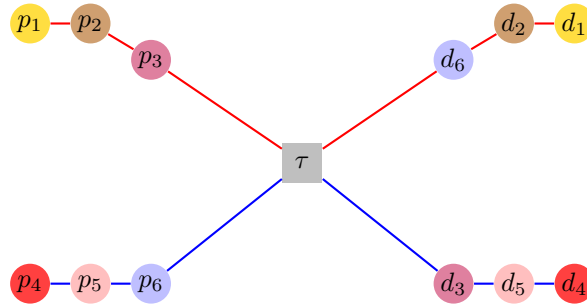


Figure 4.2 – Solution avec transfert des requêtes 3 et 6

Le problème de collectes et livraisons avec transferts a donné lieu à un nombre assez modeste de travaux, et n'a été formellement modélisé qu'en 2010 par Cortés *et al.* (2010).

Le tableau 4.1 synthétise les principales références bibliographiques sur ce problème. Seules les références les plus récentes considèrent à la fois les fenêtres horaires et les contraintes de capacité. Les méthodes de résolution exacte ne parviennent à résoudre que de petites instances. Des méthodes de type branch-and-cut sont utilisées par Kerivin *et al.* (2008) et Cortés *et al.* (2010). La contribution principale de l'article de Rais *et al.* (2014) est de proposer une nouvelle modélisation.

Nos travaux s'articulent selon trois axes de recherche. Tout d'abord, nous avons souhaité travailler sur une méthode de résolution exacte pour le PDPT, avec comme objectif de résoudre des instances de taille réaliste. Devant la difficulté du problème, nous avons défini une variante du PDPT, appelée Pickup and Delivery Problem with Shuttle routes (PDPS), résolue par une méthode de type branch-and-cut-and-price. Ce travail, réalisée en collaboration avec Stefan Ropke (DTU, Copenhague) est décrit en section 4.2. Ensuite, nous avons implémenté une heuristique de type ALNS pour le PDPT. Ce travail est décrit en section 4.3. Le modèle a finalement été étendu pour prendre en compte des contraintes de temps de trajet maximal pour les passagers, donnant lieu à un problème inédit : le DARP avec transferts (DARPT) (section 4.4).

4.2 Le problème de collectes et livraisons avec navettes (PDPS)

Le problème de collectes et livraisons avec navettes (Masson *et al.*, 2014b) est une extension du PDP qui autorise qu'un passager soit déposé en un autre point que sa destination. Un système de trajets directs, appelés navettes, entre les points de destination, permet alors d'amener chaque passager à bon port. Cette modélisation est pertinente dans le cas où le nombre de points de destination est faible comparé au nombre de points de collecte. La Figure 4.3 illustre une solution du PDPS avec trois points de destination.

Articles	Fenêtres horaires	Capacité	Tps. Trajet Max.	Nb. requêtes	Méthodes
Shang et Cuff (1996)	✓	✗	✗	300	Heur. Constructive
Mitrović-Minić et Laporte (2006)	✓	✗	✗	100	Rech. Locale
Lin (2008)	✓	✗	✗	100	Heur. B&C
Kerivin <i>et al.</i> (2008)	✗	✓	✗	15	Exacte
Cortés <i>et al.</i> (2010)	✓	✓	✗	6	Exacte
Qu et Bard (2012)	✓	✓	✗	25	GRASP+ALNS
Rais <i>et al.</i> (2014)	✓	✓	✗	7	Exacte
Positionnement de nos travaux					
Masson <i>et al.</i> (2014b) (PDPS)	✓	✓	✗	87	Exacte
Masson <i>et al.</i> (2013a) (PDPT)	✓	✓	✗	193	ALNS
Masson <i>et al.</i> (2014a) (DARPT)	✓	✓	✓	193	ALNS

Tableau 4.1 – Etat de l’art sur les problèmes avec transfert

On suppose que toutes les requêtes ayant un même point de livraison partagent la même fenêtre horaire sur ce point et qu’un véhicule ne peut pas visiter plus de deux points de livraison. Ces hypothèses sont adaptées aux établissements médico-sociaux, qui interdisent aux personnes d’arriver trop longtemps avant l’ouverture. En raison de ces fenêtres horaires serrées, les véhicules ne peuvent pas visiter plus de 2 établissements.

On définit ainsi les *tournées de collecte*, qui visitent plusieurs points de collecte puis un point de livraison et les *navettes*. Les tournées de collecte sont représentées en traits plein sur la Figure 4.3; les navettes sont représentées en traits hachurés. En pratique, rien n’interdit à un véhicule d’enchaîner plusieurs tournées (collecte+navette ou navette+navette), mais ces regroupements de trajets sont réalisés a posteriori à partir d’une solution du PDPS.

4.2.1 Modélisations basées sur les chemins

Les modélisations de type partitionnement (*set partitionning*) constituent une approche classique pour résoudre les problèmes de tournées de véhicules avec des méthodes de génération de colonnes. Une solution du PDPS est un ensemble de routes indépendantes correspondant au trajet de chaque véhicule. Une route est une séquence ordonnée de sommets entre un dépôt initial fictif et un dépôt final fictif, qui ne passe que par des arcs existants, respecte la capacité des véhicules et les contraintes de fenêtres horaires. Le coût d’une route est la somme des coûts des arcs qui le composent.

Dans le cas des tournées avec transfert, les contraintes de synchronisation aux points de transfert font tomber l’hypothèse d’indépendance des routes. Afin de rendre les routes indépendantes les unes des autres, et donc pouvoir proposer un modèle de partitionnement, nous proposons deux approches :

- modélisation **DT** : dupliquer chaque point de transfert autant de fois qu’il y a de requêtes dans le problème.
- modélisation **DC** : dupliquer chaque point de collecte autant de fois qu’il y a de points de transfert dans le problème.

De même, nous proposons deux façons de modéliser les navettes. Dans la première, on ne liste pas explicitement les requêtes transportées par navette. Comme la flotte est homogène, connaître le nombre de navettes entre chaque paire de points de livraison suffit à calculer le coût d’une solution. On parle alors de modélisation *agrégée* (notée **Ag**).

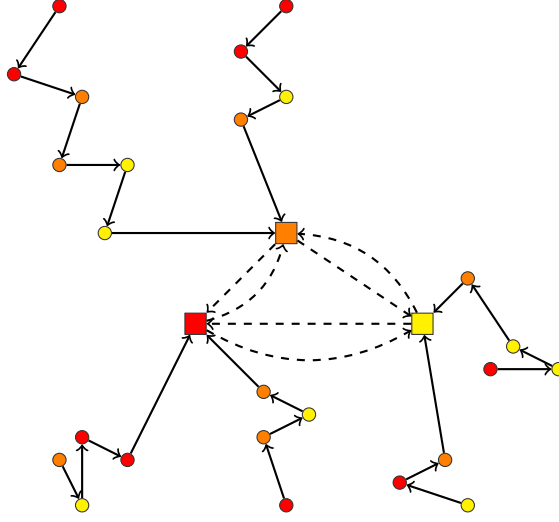


Figure 4.3 – Une solution du PDPS

La deuxième modélisation décrit explicitement la composition de chaque navette. Cela requiert de créer les véhicules navette a priori. L'affectation des requêtes à ces véhicules est réalisée par l'algorithme de résolution. Cette modélisation est appelée *explicite* et notée **Ex**.

On note P l'ensemble des points de collecte et T l'ensemble des points de destination, servant aussi de point de transfert. Ω désigne l'ensemble de toutes les routes possibles et Ω' l'ensemble des routes de collecte. Le coût d'une route $r \in \Omega$ est noté c_r . Dans le cadre des modélisations **DC**, on définit les données $a_p^{i,\tau}$ valant 1 si et seulement si les sommets $i \in P$ et $\tau \in T$ sont tous les deux servis par la route $p \in \Omega$, 0 sinon. $v_{\tau,\tau'}$ est le coût associé à une navette entre les points de livraison $\tau \in T$ et $\tau' \in T$. Enfin, on note R_τ l'ensemble des requêtes dont la destination se situe à l'emplacement $\tau \in T$.

Les variables de décision sont les suivantes : $\omega_r = 1$ si la route $r \in \Omega$ est utilisée dans la solution, 0 sinon. $y_{\tau,\tau'}$ est une variable entière représentant le nombre de navettes effectuées entre deux points de livraison $\tau \in T$ et $\tau' \in T$. On considère $y_{\tau,\tau} = 0$ quel que soit $\tau \in T$.

La combinaison des deux approches **DC** et **DT** et des deux niveaux d'agrégation **Ag** et **Ex** donne lieu à 4 modèles mathématiques notés respectivement **(DT-Ag)**, **(DT-Ex)**, **(DC-Ag)** et **(DC-Ex)**.

Le modèle **(DC-Ag)** s'écrit :

$$\min \quad \sum_{p \in \Omega'} c_p \omega_p + \sum_{\tau \in T, \tau' \in T} v_{\tau,\tau'} y_{\tau,\tau'} \quad (4.1)$$

s.c.

$$\sum_{p \in \Omega} \sum_{\tau \in T} a_p^{i,\tau} \omega_p = 1 \quad \forall i \in P \quad (4.2)$$

$$\sum_{p \in \Omega, r \in R_\tau} a_p^{r,\tau} \omega_p \leq Q y_{\tau,\tau'} \quad \forall \tau \in T, \tau' \in T, \tau' \neq \tau \quad (4.3)$$

$$\omega_p \in \{0, 1\} \quad \forall p \in \Omega', \quad (4.4)$$

$$y_{\tau,\tau'} \in \mathbb{Z}^+ \quad \forall \tau \in T, \tau' \in T. \quad (4.5)$$

La fonction objectif (4.1) est composée de deux termes représentant le coût des tournées de collecte et le coût des navettes. Les contraintes (4.2) imposent que chaque requête soit acheminée vers un

point de livraison et un seul. Les contraintes (4.3) assurent que le nombre de véhicules effectuant une navette entre les points de livraison τ et τ' soit suffisant pour transporter toutes les requêtes de $\tau \in T$ à $\tau' \in T$.

Le détail des quatre variantes (**DC-Ag**), (**DC-Ex**), (**DT-Ag**) et (**DT-Ex**) est présenté dans la thèse de Renaud Masson (Masson, 2012).

4.2.2 Branch-and-Cut-and-Price

Le PDPS est résolu par une méthode de séparation et évaluation. Chaque noeud de l'arbre de recherche évalue une borne inférieure correspondant à la valeur optimale de la relaxation linéaire du PDPS en ce noeud. La relaxation linéaire sur les noeuds de l'arbre de recherche est calculée par une méthode de génération de colonnes. Elle est renforcée par l'ajout d'inégalités valides. On utilise donc une méthode de type Branch-and-Cut-and-Price.

Dans les quatre modèles proposés, les coûts réduits représentent la somme des coûts des arcs empruntés plus la somme des coûts associés aux noeuds visités. La recherche de chemins ayant le plus petit coût réduit revient dans tous les cas à résoudre un plus court chemin entre les noeuds o et o' respectant des contraintes relatives aux capacités des véhicules et aux fenêtres horaires. Ce sous-problème appartient à la famille des problèmes de plus court chemin élémentaire avec contraintes de ressources (ESPPRC : Elementary Shortest Path Problem with Resource Constraints). L'ESPPRC est résolu par un algorithme de type *label correcting* qui construit des chemins partiels et s'appuie sur le principe d'étiquetage et d'extension de chemins partiels (Feillet *et al.*, 2004).

La relaxation linéaire est renforcée avec trois familles d'inégalités valides : les *rounded capacity inequalities* (Lysgaard, 2003), les *two-path inequalities* (Kohl *et al.*, 1999) et les *subset-row inequalities* (Jepsen *et al.*, 2008). Nous avons effectué des tests numériques pour évaluer l'impact de ces inégalités. Leur ajout permet de réduire la taille de l'arbre de recherche, mais parfois au prix d'un effort de calcul pour séparer ces coupes. Au final, nous conservons les trois familles d'inégalités valides au noeud racine, et seulement les *rounded capacity inequalities* par la suite, ce qui permet de diviser le temps de calcul par un facteur 3 par rapport à un branch-and-price.

Nous utilisons les trois règles de branchement suivantes :

1. Comme la variable ω_p associée à l'utilisation des colonnes $p \in \Omega$ est binaire, la somme de ces variables doit être entière. Si ce n'est pas le cas, un branchement sur la valeur de cette somme est effectué.
2. On branche sur le nombre de navettes, autrement dit sur les variables y dans les modèles **Ag** et sur les variables ω dans les modèles **Ex**.
3. On branche sur l'utilisation ou non d'un arc : si la somme des variables ω_p traversant un arc donné n'est pas entière, deux branches sont créées : l'une impose l'utilisation de l'arc, alors que l'autre l'interdit. On branche en priorité sur les arcs dont le flot est proche de 0.5.

4.2.3 Résultats numériques

L'algorithme a été testé d'une part sur un jeu de 81 instances générées (27 instances avec 25, 50 et 75 requêtes). Le temps de calcul a été limité à 1 heure. Le tableau 4.2 synthétise les résultats numériques, en indiquant le nombre d'instances résolues optimalement pour chaque type de modèle. On observe que toutes les instances à 25 requêtes, et environ la moitié des instances à 75 requêtes peuvent être résolues.

Le Tableau 4.3 montre les moyennes des bornes inférieures obtenues avec chacune des modélisations. La différence entre les modèles est minimale sur les instances à 50 requêtes. En revanche, les modèles de type **DC** permettent d'obtenir de meilleures bornes pour les instances à 75 requêtes.

Nous avons ensuite appliqué l'algorithme aux données réelles fournies par l'ADAPEI44. Le tableau 4.4 résume les résultats. 4 des 5 instances ont pu être résolues avec un saut d'optimalité inférieur à

Requêtes	Modèle			
	DT-Ex	DT-Ag	DC-Ex	DC-Ag
25	27/27	27/27	27/27	27/27
50	23/27	20/27	24/27	23/27
75	15/27	13/27	14/27	13/27
Total	65/81	60/81	65/81	63/81

Tableau 4.2 – Nombre d’instances résolues optimalement

Requêtes	Modèle			
	DT-Ex	DT-Ag	DC-Ex	DC-Ag
25	465,41	465,41	465,41	465,41
50	712,63	711,39	712,16	712,40
75	921,95	963,45	986,59	986,57
Moyenne	700,00	713,42	721,39	721,46

Tableau 4.3 – Moyennes des bornes inférieures

1%. L’instance ADAPEI-87-2 (87 requêtes, 2 établissements) est même résolue optimalement. En revanche, pour l’instance ADAPEI-193-5 (193 requêtes, 5 établissements), l’écart entre les bornes inférieure et supérieure est encore de 4.19% après une heure de calcul.

Instances	DC-Ag		
	LB	UB	Ecart
ADAPEI-81-2	433.69	436.22	0.58%
ADAPEI-84-2	828.83	835.77	0.83%
ADAPEI-87-2	461.40	461.40	0.00%
ADAPEI-109-2	607.93	614.92	1.14%
ADAPEI-193-5	1098.15	1146.12	4.19%

Tableau 4.4 – Résultats du Branch-and-cut-and-price sur les données ADAPEI44

Ce travail constitue un premier pas vers la résolution exacte d’instances de taille réelle du PDPT. Il s’appuie sur des hypothèses fortes qui permettent de structurer le problème et de pouvoir appliquer une approche de génération de colonnes. L’interconnexion des routes aux points de transfert vient compliquer la formulation du problème maître et oblige à dupliquer soit les points de transfert, soit les points de collecte, ce qui contribue à agrandir la taille du problème maître.

Les hypothèses prises ici sont toutefois assez réalistes pour l’application de ce problème au transport de personnes handicapées. La première extension possible serait de considérer des véhicules de taille différente pour les navettes et pour les tournées de collecte.

4.3 Le problème de collectes et livraisons avec transferts (PDPT)

Nous abordons maintenant le problème de collectes et livraisons avec transferts (PDPT) à proprement parler. Nous en rappelons les principales hypothèses :

- n requêtes doivent être transportées d’un point de collecte vers un point de livraison, en satisfaisant des contraintes horaires.

- On dispose d’une flotte homogène de véhicule, de capacité limitée.
- Les requêtes peuvent être transférées d’un véhicule à un autre en une liste prédéterminée de points appelés *points de transfert*.
- Sans perte de généralité, on considère que la liste des points de transfert coïncide avec les points de livraisons, et que le nombre de points de livraison est largement inférieur au nombre de points de collecte.
- Les requêtes sont transférées au plus une fois.

L’objectif est de construire un ensemble de tournées de coût minimal satisfaisant l’ensemble des requêtes.

4.3.1 Une recherche adaptative à voisinage large (ALNS) pour le PDPT

Devant le succès rencontré par la recherche adaptative à voisinage large (ALNS) pour une grande variété de problèmes de tournées (Pisinger et Ropke, 2007) et en particulier pour le problème de collectes et livraisons avec fenêtres horaires (PDPTW) (Ropke et Pisinger, 2006), nous avons adapté cette méthode au PDPT (Masson *et al.*, 2013a). Notre approche a été d’utiliser des opérateurs déjà décrits par (Ropke et Pisinger, 2006) pour la résolution du PDPTW et de les enrichir par de nouveaux opérateurs définis spécifiquement pour les problèmes avec transferts.

Nous avons donc réutilisé les opérateurs suivants :

- *random removal* : retrait aléatoire de requêtes,
- *worst removal* : retrait en priorité des requêtes contribuant le plus à l’augmentation de la fonction objectif,
- *related Removal* : retrait de requêtes semblables. La mesure de similarité entre deux requêtes est basée sur la distance entre leurs points de collecte, la distance entre leurs points de livraison, les charges associées à ces requêtes, la différence entre leurs fenêtres de temps et le fait qu’elles soient servies ou non par la même tournée dans la solution courante.
- *best insertion* : on insère en priorité les requêtes non planifiées ayant le plus petit coût d’insertion dans la solution courante,
- *k-regrets* avec des valeurs de k allant de 2 à 5.

Nous avons ensuite proposé des opérateurs construits spécifiquement pour les problèmes avec transferts :

- *Retrait d’un point de transfert* : inspiré d’un opérateur proposé par Lapiere *et al.* (2004), cet opérateur retire toutes les requêtes utilisant un point de transfert donné, et les re-route par ailleurs.
- *Retrait de clusters* : cet opérateur vise à retirer simultanément des requêtes pouvant être routées efficacement via un point de transfert commun. Cela est le cas si soit les points de collecte, soit les points de livraison forment un cluster.
- *History Removal* : retrait basé sur l’historique. Cet opérateur vise à retirer les requêtes qui semblent être mal positionnées dans la solution courante, au regard des meilleures solutions trouvées aux itérations précédentes. Nous avons adapté l’opérateur décrit par Pisinger et Ropke (2007) pour tenir compte des requêtes transférées.
- *Meilleure insertion avec transfert* : il s’agit d’une adaptation du voisinage proposé par Mitrović-Minić et Laporte (2006), qui cherche la meilleure insertion possible d’une requête en passant par un point de transfert. Nous n’avons pas implémenté l’opérateur proposé par Qu et Bard (2012), potentiellement plus performant mais de complexité $O(|T|n^4)$ au lieu de $O(|T|n^2)$.
- *Transfert d’abord* : cet opérateur force le regroupement de plusieurs requêtes en un point de transfert, et permet de sortir de certains optima locaux.
- *k-regret avec transfert* : nous généralisons la meilleure insertion avec transfert, en utilisant la notion de regret.

4.3.2 Réalisabilité d'une insertion de requête dans une solution

En raison de l'interconnexion des routes aux points de transfert, vérifier s'il est possible d'insérer une requête dans une solution courante n'est plus un problème trivial. Tout d'abord, l'insertion d'une requête revient à insérer deux sommets (montée et descente) dans une même route ou deux routes distinctes. Étant donné le très grand nombre de tests d'insertion réalisés tout au long de l'exécution de l'ALNS, il est capital que cette opération puisse se faire en temps constant. Nous avons étendu la notion de *forward time slack* (Savelsbergh, 1992) afin de vérifier la réalisabilité en temps constant (Masson *et al.*, 2013b). Obtenir cette complexité nécessite néanmoins un pré-traitement en temps quadratique en fonction du nombre de sommets, à chaque fois qu'une insertion est réalisée. En pratique, lorsque le nombre d'insertions réalisées est faible par rapport au nombre d'insertions testées, ce pré-traitement n'est pas pénalisant.

De plus, chaque transfert de requête implique une relation de précédence : si une requête est transférée du véhicule v_1 vers le véhicule v_2 , alors v_1 doit arriver au point de transfert avant que v_2 n'en reparte. Ces relations de précédence peuvent être représentées par un graphe dans lequel les arcs représentent les liens de précédence entre sommets, et où chaque point de transfert est représenté par deux sommets : le premier correspondant à l'arrivée du véhicule v_1 et l'autre au départ du véhicule v_2 .

La figure 4.4 illustre un tel cas. Le transfert de la requête 1 au point de transfert τ_1 est représenté par un arc entre le sommet τ_1^- et le sommet τ_1^+ . Cet arc connecte les routes 1 et 2. Insérer la requête 2 entre p_2 et d_2 nécessiterait de passer par les sommets τ_2^- et le sommet τ_2^+ , ce qui créerait un cycle dans le graphe. Or, la présence d'un cycle exprime une relation de précédence entre un sommet et lui-même, ce qui est inconsistent.

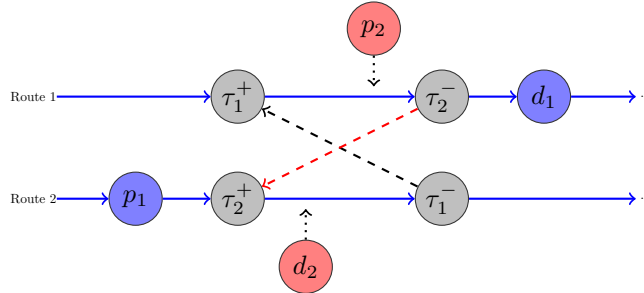


Figure 4.4 – Exemple d'une insertion de requête non réalisable

Tester la réalisabilité d'une insertion de requête nécessite donc de rechercher la présence de tels cycles dans le graphe de précédence. Cette vérification peut être réalisée en temps constant pourvu que le graphe de précédence soit stocké en mémoire avec une structure de donnée adéquate (par exemple une matrice d'incidence sommets-sommets).

4.3.3 Résultats numériques

Les performances de l'algorithme ALNS ont été testées sur trois types de données : les instances générées par Mitrović-Minić et Laporte (2006), les données fournies par l'ADAPEI44 et celles fournies par le Conseil Général de Loire Atlantique (CG44). L'algorithme a été stoppé après 25000 itérations, avec entre 10% et 20% de requêtes détruites puis reconstruites à chaque itération.

Nous avons tout d'abord souhaité tester la capacité de l'algorithme à produire de bonnes solutions. Ne connaissant pas la valeur des solutions optimales du PDPT, il est difficile d'évaluer la performance de l'ALNS en terme de fonction objectif. En revanche, la résolution du PDPS fournit des solutions optimales ou d'excellentes bornes inférieures pour un problème très proche. Nous avons donc testé les performances de l'ALNS en l'adaptant pour résoudre le PDPS. Le Tableau

4.5 montre une comparaison entre la borne inférieure fournie par l'algorithme de branch-and-cut-and-price et la borne supérieure trouvée par l'ALNS¹. Le saut d'optimalité est inférieur à 1% pour 4 des 5 instances, mais vaut 5.51% pour l'instance la plus difficile. Le branch-and-cut-and-price ayant lui même un saut d'optimalité de plus de 4% , il est difficile de conclure quoi que ce soit pour cette instance.

Instances	B&C&P	ALNS	Ecart
	Borne inf.	Borne Sup.	
ADAPEI-81-2	428.31	430.72	0.81%
ADAPEI-84-2	768.79	775.00	0.56%
ADAPEI-87-2	456.82	457.83	0.22%
ADAPEI-109-2	598.57	603.95	0.90%
ADAPEI-193-5	1067.85	1126.65	5.51%

Tableau 4.5 – Écart de l'ALNS à la borne inférieure du PDPS

Dans un deuxième temps, nous avons comparé les résultats de l'ALNS avec ceux obtenus par la recherche locale proposée par Mitrović-Minić et Laporte (2006). Parmi les 4320 instances générées par Mitrović-Minić et Laporte (2006), nous en avons sélectionné 780 issues de 26 classes différentes, représentatives de l'ensemble des instances. En moyenne les résultats de l'ALNS améliorent de 7.04% ceux de la recherche locale sur les instances avec fenêtres de temps longues, et de 14.41% sur les instances avec fenêtres de temps courtes. Ces moyennes cachent de grandes variations : pour 2 des 26 classes, nous obtenons de moins bons résultats (-1.61% et -0.46%), mais sur les autres classes l'écart peut monter jusqu'à 21% en faveur de l'ALNS.

Par la suite, nous avons souhaité évaluer l'intérêt de recourir aux transferts. Pour ce faire, nous avons tout simplement comparé les résultats du PDPT et ceux du PDP. Sur les instances de Mitrović-Minić et Laporte (2006), le recours aux transferts permet de diminuer la distance totale parcourue entre 1.46% et 28.42% selon les configurations. Sans surprise le cas le plus défavorable correspond à celui où les requêtes sont très dispersées avec un seul point de transfert situé au centre de la zone. Dans le cas le plus favorable, les requêtes sont plus regroupées et plusieurs points de transfert sont facilement accessibles pour chaque requête.

Sur les instances ADAPEI 44 et CG44, le gain apporté par les transfert est compris entre 0.75% et 9.23%. Le pire cas correspond à une instance CG44 avec seulement 55 requêtes pour 16 point de destination. Il n'existe alors pas un potentiel suffisant de massification des flux pour rentabiliser le détour par un point de transfert.

Cette notion de détour pour rejoindre un point de transfert explique à elle seule la différence de gain entre les instances de Mitrović-Minić et Laporte (2006) et les instances réelles. Dans le premier cas, la distance entre deux points est calculée avec la distance de Manhattan. Une conséquence heureuse de cette métrique est qu'entre toute paire de points A et B , il existe une infinité de plus courts chemins, tous contenus dans un rectangle de diagonale AB . Si un point de transfert est situé dans ce rectangle, il est atteignable sans aucun détour par rapport au plus court chemin. En revanche, pour les instances réelles nous avons calculé les itinéraires avec le logiciel Microsoft MapPoint, en suivant les axes routiers. Déposer une requête sur un point de transfert occasionne donc un détour pénalisant.

4.4 Dial a Ride Problem avec Transfert (DARPT)

La description du DARPT diffère de celle du PDPT par seulement deux nouvelles familles de contraintes : on impose à chaque requête un temps de trajet maximal et à chaque tournée une

¹. Les hypothèses prises en compte ici et dans le Tableau 4.4 ne sont pas exactement les mêmes, il est donc normal que les bornes inférieures (colonne 2) diffèrent.

limite de durée. L'objectif est également de construire un ensemble de tournées pour satisfaire l'ensemble des requêtes à coût minimal.

Le principe de résolution réutilise la métaheuristique ALNS déjà employée pour le PDPT. Malgré les similitudes entre les deux problèmes, le DARPT est beaucoup plus difficile à résoudre, et ce pour une raison majeure : lorsqu'on insère une requête dans une tournée, il est nécessaire de vérifier le respect des contraintes temporelles. Or les tournées sont interconnectées. Ainsi, l'insertion d'une requête dans une tournée peut entraîner la non réalisabilité d'une autre tournée. Cela est illustré par les figures 4.5 et 4.6. La Figure 4.5 montre deux tournées sur lesquelles on a représenté 2 requêtes.

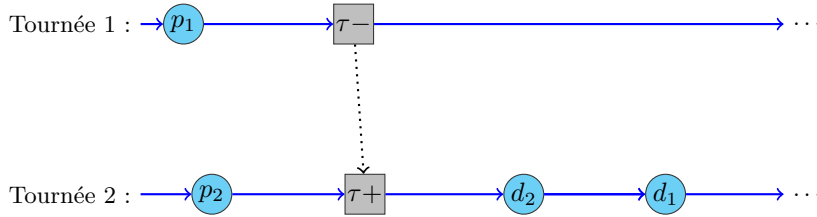


Figure 4.5 – Avant insertion d'une requête

La requête 1, matérialisée par les sommets p_1 et d_1 , est transférée de la tournée 1 à la tournée 2 au point de transfert représenté par deux sommets τ^- et τ^+ . La requête 2 est intégralement servie par la tournée 2.

La Figure 4.6 montre l'effet que peut avoir une insertion de requête sur la tournée 1 avant p_1 . Supposons que cette insertion retarde la date de passage en p_1 . Cela retarde automatiquement l'heure du transfert (arc de τ^- à τ^+) ainsi que tous les sommets de la tournée 2 situés après τ^+ . Notamment, le passage au point de livraison d_2 a été retardé alors que le passage en p_2 n'a pas été modifié. Le temps de trajet de la requête 2 a donc été augmenté et est donc susceptible de violer la contrainte de temps de trajet maximal.

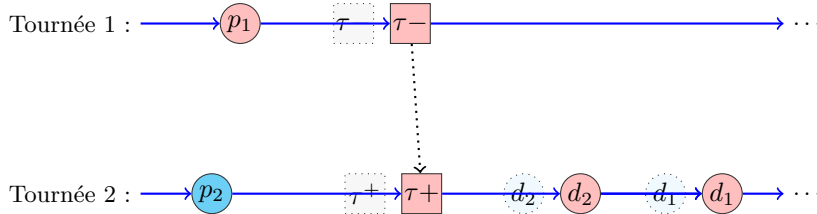


Figure 4.6 – Après insertion d'une requête sur la route 1 avant $P1$

Vérifier la réalisabilité d'une insertion de requête dans une solution courante revient donc à déterminer une heure de service sur chaque sommet, qui respecte à la fois les contraintes de précedence, de fenêtres horaires et de temps de trajet maximal. Le problème de vérification des horaires est équivalent à un cas particulier du *Problème de satisfaction de contraintes temporelles* (Temporal Constraint Satisfaction Problem : TCSP) (Dechter *et al.*, 1991). Ce cas particulier est le *Problème Temporel Simple* (Simple Temporal Problem : STP). Le STP peut être représenté par un graphe orienté, appelé *graphe de distance*, où chaque sommet représente une variable temporelle et chaque arc représente une contrainte entre deux variables. Un STP est cohérent si et seulement si le graphe associé ne contient pas de circuit de longueur positive (Shostak, 1981).

Plusieurs algorithmes existent pour chercher de tels circuits dans un graphe orienté. Nous utilisons l'algorithme BFCT (Tarjan, 1981), qui est une variation de l'algorithme de Bellman-Ford. Cet algorithme a une complexité $O(nm)$. Si aucun circuit de longueur positive n'est trouvé, il génère une planification au plus tôt.

Malheureusement, évaluer toutes les positions d’insertion de toutes les requêtes non planifiées, et ce à chaque itération de l’ALNS, représenterait un temps de calcul dissuasif. Afin de s’épargner de trop nombreux appels à BFCT, nous avons donc mis en place des filtres, sous la forme de conditions nécessaires et de conditions suffisantes de réalisabilité faciles à vérifier.

- Condition nécessaire 1 (CN1) : on teste une insertion de requête en ignorant les contraintes de temps de trajet maximum. Cette condition nécessaire peut être vérifiée en temps constant.
- Condition suffisante 1 (CS1) : on tente d’insérer une requête sans modifier les horaires sur la route. Cette condition suffisante peut être vérifiée en temps constant.
- Condition nécessaire 2 (CN2) : on relaxe les contraintes de précédence aux points de transfert. Chaque route devient ainsi indépendante. On peut utiliser l’algorithme proposé par Cordeau et Laporte (2003), de complexité $O(n^2)$ (où n est le nombre de sommets dans la route) pour fixer les heures de service d’une tournée.
- Condition suffisante 2 (CS2) : replanification basée sur l’algorithme de Cordeau et Laporte (2003). Si CN2 est satisfaite, les requêtes non transférées dans la solution courante ont un temps de trajet cohérent. Une condition suffisante est obtenue si l’horaire proposé dans la CN2 propose des heures de service réalisables pour les requêtes transférées. Cette vérification se fait en $O(n^2)$.

Notons que nous utilisons ces conditions nécessaires et conditions suffisantes par ordre croissant de complexité. Le processus est illustré sur la Figure 4.7.

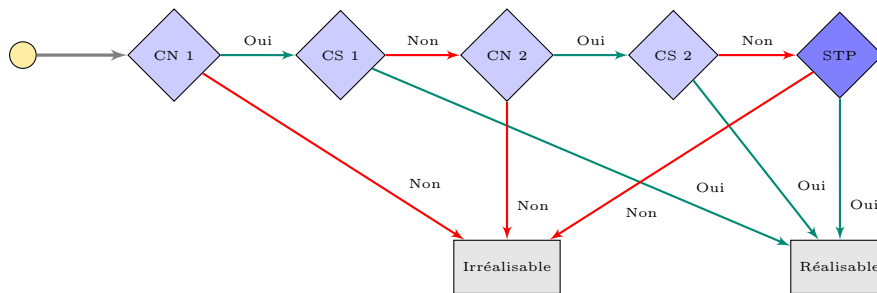


Figure 4.7 – Utilisation des conditions nécessaires et de conditions suffisantes

Nous avons mesuré l’impact de ce filtrage en comparant trois approches : utiliser systématiquement l’algorithme BFCT, utiliser uniquement les conditions nécessaires et utiliser toutes les conditions nécessaires et suffisantes. Le tableau 4.4 montre le temps d’exécution de l’algorithme ALNS dans les trois scénarios, ainsi que le pourcentage d’insertions réalisables. On y voit que les conditions nécessaires contribuent surtout à diminuer le temps de calcul. On observe aussi que ce filtrage est plus efficace quand le pourcentage d’insertions réalisables est faible.

La modélisation complète du problème STP et sa résolution ont été publiées dans les actes de la conférence CPAIOR 2012 (Masson *et al.*, 2012). Nous avons également évalué le gain apporté par l’utilisation des points de transfert sur les mêmes instances réelles que pour le PDPT. Les gains apportés par les transferts varient entre 0.97% et 8.24%. Le temps de calcul augmente là aussi largement si on considère les transferts. Ces tests numériques sont détaillés dans Masson *et al.* (2014a).

4.5 Conclusions

Ce chapitre décrit nos travaux sur l’optimisation de plusieurs problèmes de transport à la demande avec transfert de passagers. Nous avons tout d’abord présenté la résolution d’une variante très structurée du PDPT avec une méthode de branch-and-cut-and-price. Cette approche a permis la résolution exacte en une heure de calcul de modèles allant jusqu’à 87 requêtes.

Instances	CPU (en min)			% d'insertions réalisables
	STP	CN1 + CN2	CN1 + CS1 + CN2 + CS2	
ADAPEI-81-2	7.2	1.8	1.4	1.3%
ADAPEI-84-2	2.4	1.0	1.0	2.8%
ADAPEI-87-2	5.3	2.2	1.6	2.4%
ADAPEI-109-2	33.0	4.1	3.1	1.1%
ADAPEI-193-5	254.2	17.1	15.0	1.2%
CG44-55-16	0.9	0.9	0.9	53.0%
CG44-66-13	1.0	1.0	1.0	39.8%
CG44-84-21	2.1	2.0	2.0	31.9%
CG44-84-33	3.4	2.8	2.7	23.1%
CG44-106-24	3.9	3.3	3.2	20.7%

Tableau 4.6 – Impact des conditions nécessaires et conditions suffisantes

Nous avons ensuite décrit les résultats de l'algorithme ALNS pour le PDPT, qui permet de traiter des instances de taille supérieure (jusqu'à 193 requêtes). Un point clé de cette résolution est de pouvoir tester en temps constant la réalisabilité d'une insertion de requête dans une solution courante. Cela est rendu possible grâce à une adaptation de la notion de forward time slack.

Le dernier problème traité est le problème inédit du DARP avec transfert. Le point clé pour résoudre ce problème en un temps de calcul acceptable est la capacité à tester rapidement la réalisabilité d'une insertion, surtout du point de vue des temps de trajets. Nous avons décrit un ensemble de conditions nécessaires et de conditions suffisantes qui permettent d'accélérer ce test de réalisabilité.

Ces travaux ont permis de faire progresser les méthodes de résolution des problèmes avec transfert. Nous avons identifié plusieurs pistes de recherche et champs d'application possibles de ces travaux. Tout d'abord, nous n'avons résolu de manière exacte qu'une variante simplifiée du PDPT. La principale difficulté d'une approche de type branch-and-price pour le PDPT est que le sous-problème à résoudre risque d'être beaucoup plus compliqué que l'ESPPRC décrit dans la section 4.2.

Ensuite, nous avons mesuré l'impact des transferts en comparant les fonctions objectif des problèmes avec ou sans points de transferts. Or, dans l'introduction de sa thèse, Renaud Masson montre que le pourcentage de gain dû aux transferts dépend fortement des requêtes qui ne sont pas transférée. Cet indicateur peut donc être modifié artificiellement en ajoutant ou retirant des requêtes non transférables. D'autres facteurs, comme le nombre de requêtes transférées et l'impact des ruptures de charges et des risques opérationnels liés à l'interconnexion des routes, sont à prendre en compte dans l'évaluation d'une solution.

Lorsqu'une organisation souhaite évaluer la pertinence de recourir aux transferts, il n'y a aujourd'hui pas d'autre moyen que de résoudre le problème avec et sans transfert, et de comparer les résultats. Il serait intéressant d'évaluer a priori l'intérêt de recourir aux transferts. Cela nécessite de connaître les caractéristiques des instances qui se prêtent bien à une utilisation de points de transfert. La répartition géographique des requêtes et des points de transfert, mais aussi la capacité des véhicules et les fenêtres horaires jouent un rôle qu'il reste à analyser.

Dans les applications décrites dans nos travaux, il était naturel de localiser les points de transfert aux points de destination des requêtes. Mais dans d'autres contextes (logistique urbaine, systèmes de distribution multi-niveaux pour la messagerie ou la presse quotidienne), la question de la localisation des points de transfert se pose très rapidement. Appliquer au PDPT une étude semblable à celle réalisée par Crainic *et al.* (2010) pour le VRP à 2 échelons permettrait d'aller vers une meilleure compréhension de ce problème.

Chapitre 5

Projets de recherche actuels

Ce chapitre décrit des travaux de recherche actuels n'ayant pas encore été publiés. Chaque section présente brièvement un projet en cours. À travers ces descriptions de projets, on voit se tisser des liens avec les travaux décrits dans les chapitres précédents. Ces liens peuvent provenir par exemple de l'utilisation des mêmes techniques de résolution ou de la présence des mêmes caractéristiques dans les problèmes étudiés.

5.1 Recherche à voisinage large pour un problème de localisation

Parmi les multiples méthodes utilisées pour résoudre les problèmes de localisation dans les chaînes logistiques, l'algorithme LNS, largement évoqué dans les chapitres précédents, n'a quasiment jamais été utilisé. Copado-Méndez *et al.* (2013) identifient pourtant un avantage lié à cette méthode : les opérateurs de destruction et de reconstruction, largement dépendants du modèle à résoudre, offrent une grande flexibilité. Ainsi, le LNS peut se présenter comme un cadre général dans lequel on définit une collection d'opérateurs qu'on mobilise ou adapte en fonction des attributs du modèle à résoudre.

En revanche, plusieurs difficultés nouvelles se font jour. Tout d'abord, il faut considérer séparément les variables binaires et les variables continues. Une approche possible est de fixer les variables binaires de localisation au moyen des opérateurs de l'algorithme LNS, et les variables continues de flux en résolvant un sous-problème à chaque itération. Ensuite, les caractéristiques de la solution optimale ne sont pas connues a priori. Dans les problèmes de tournées de véhicules, le nombre de clients à visiter est fixe, de même pour le nombre de tâches dans les problèmes d'ordonnancement. En localisation, il est impossible de savoir à l'avance le nombre de sites actifs dans une solution optimale de problème (sauf bien sûr pour le problème p -médiann). Dans un réseau multi-échelons, on appelle *configuration* le vecteur qui indique le nombre de sites actifs à chaque échelon. Plus le réseau a d'échelons et de sites candidats, plus le nombre de configurations possibles est élevé. Une bonne méthode heuristique doit pourtant assurer une bonne couverture de ces configurations pour trouver la solution optimale. Enfin, plusieurs attributs du problème, tels que le choix de modes de transport ou de technologies de production sont modélisés par des variables binaires et qui viennent compliquer les modèles. Ces variables peuvent être traitées par les opérateurs de destruction, mais aussi par des heuristiques spécifiques intégrées dans l'algorithme LNS.

Les travaux de thèse de Majid Eskandarpour (oct. 2011–déc.2014), encadrés avec Pierre Dejax (École des Mines de Nantes) et Joe Miemczyk (Audencia, Nantes) portent sur l'implémentation d'une méthode générique de type LNS pour des modèles riches de localisation. Les travaux

concernent notamment un modèle multi-niveaux intégrant le choix entre plusieurs modes de transport.

L'algorithme développé incorpore plusieurs heuristiques gloutonnes pour déterminer les modes de transport et l'ensemble des variables continues. Comme c'est le cas des différents opérateurs dans l'ALNS, chaque configuration est évaluée par un score. À chaque itération, une *configuration cible* est sélectionnée aléatoirement en fonction des scores de toutes les configurations. Les opérateurs de destruction et de reconstruction ont alors la charge de passer de la configuration courante à la configuration cible.

Cet algorithme a été testé sur 60 instances générées aléatoirement et ses performances comparées avec celles de Cplex 12.6. En moyenne, l'écart relatif entre Cplex et le LNS est de 1.43% en faveur de Cplex. Mais sur les instances les plus grandes (60 sites candidats, 300 clients à servir), l'heuristique LNS trouve en 15 minutes des solutions meilleures que Cplex après 3h de calcul.

Ce travail a été présenté à la conférence ILS 2014 (Eskandarpour *et al.*, 2014a) et soumis à Computers & Operations Research en Août 2014 (Eskandarpour *et al.*, 2014b). Une extension de ce travail à un modèle bi-objectif (coût et impact environnemental) consiste à intégrer le LNS dans une approche de type *multi-directional local search* (Tricoire, 2012). La suite des travaux prévoit une extension à un modèle multi-périodes.

5.2 État de l'art : problèmes de localisation et développement durable

L'idée d'écrire un état de l'art sur les modèles et méthodes de localisation intégrant les principes du développement durable s'est imposée d'elle même lorsque nous avons élaboré un plan de travail pour la thèse de Majid Eskandarpour. Plusieurs états de l'art sur les problèmes de localisation ou plus généralement sur la conception de chaînes logistique ont été publiés ces 15 dernières années dans la communauté recherche opérationnelle (Beamon, 1998; Owen et Daskin, 1998; Daskin *et al.*, 2005; Meixell et Gargeya, 2005; Sahin et Süral, 2007; Akçalı *et al.*, 2009; Melo *et al.*, 2009). Mais aucun ne parle de développement durable. Par ailleurs de multiples états de l'art sur la gestion de la chaîne logistique et le développement durable ont été publiés en sciences de gestion. Mais très peu d'entre eux mentionnent des modèles et méthodes de recherche opérationnelle.

Nous avons donc entrepris une revue de la littérature (Eskandarpour *et al.*, 2015), avec le cadre suivant : les articles retenus doivent (i) être écrits en anglais (ii) présenter un modèle d'optimisation avec des variables de localisation de sites (iii) mesurer explicitement l'impact environnemental ou social. La règle (iii) a permis notamment de filtrer de nombreux articles parlant de logistique inverse ou en boucle fermée, mais sans évaluation environnementale ou sociale.

Cette démarche a permis de recenser 87 articles, publiés dans 41 journaux différents. Un fait marquant est que 90% de ces articles ont été publiés depuis 2008. La figure 5.1 montre comment ces articles couvrent les 3 dimensions du développement durable.

Sans surprise, la grande majorité des articles procède à une évaluation économique et environnementale des décisions de localisation. Environ la moitié de ces articles s'inscrit dans une démarche d'Analyse du Cycle de Vie (ACV). Nous avons donc répertorié les principales caractéristiques de ces ACV : champ de l'étude, méthode d'évaluation de l'impact. Pour les autres articles, nous avons analysé sur quels attributs des modèles étudiés est mesuré l'impact environnemental : sites à localiser, transport, produits. La dimension sociale, plus difficile à quantifier, mais également plus difficile à caractériser en un certain nombre de critères largement partagés, est moins souvent présente dans les modèles mathématiques.

Nous passons ensuite en revue les modèles mathématiques et les méthodes de résolution, en distinguant modèles mono-objectif et modèles multi-objectifs largement majoritaires. Une première constatation est que la totalité des études entrant dans la catégorie ACV utilise des langages de

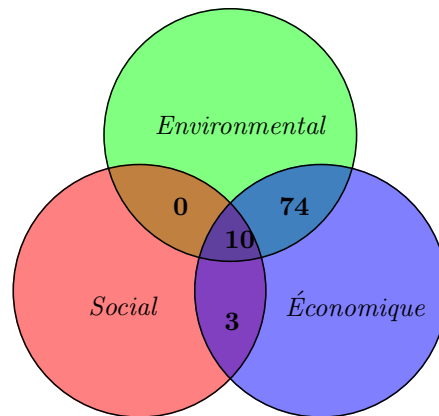


Figure 5.1 – Distribution des 87 articles de référence selon les 3 dimensions du développement durable

modélisation et des solveurs mathématiques, tandis que les autres études utilisent une panoplie beaucoup plus large de méthodes.

Il semble donc exister une dichotomie entre deux familles de chercheurs. La communauté recherche opérationnelle résout de manière parfois très fine des modèles intégrant partiellement (voire grossièrement) les principes du développement durable. A contrario, les chercheurs issus des domaines d'application (chimie, énergie, industrie, agriculture) construisent des modèles très complets mais n'utilisent que des outils standard de résolution. Une collaboration entre ces deux communautés sera nécessaire pour résoudre des modèles de grande taille ou des extensions telles que les problèmes de localisation-routage.

5.3 Projet ORLoGES

L'objectif du projet ORLoGES (Optimisation du Réseau Logistique en Génie civil, avec les aspects Economiques et Sociétaux) est de développer une suite logicielle d'aide à la décision et d'assistance à la gestion logistique, pour une entreprise de travaux publics. Le défi majeur est d'intégrer dans un même logiciel des décisions stratégiques, tactiques et opérationnelles. Ainsi, un module concerne la localisation des sites utilisés par l'entreprise pour extraire des matériaux, les entreposer et les enfouir. Un deuxième module concerne la planification tactique des opérations. Enfin, un module opérationnel optimise les transports au jour le jour et réagit aux aléas. Ce projet est coordonné par l'entreprise Luc Durand (Pruillé, 49). Il est réalisé en collaboration avec les entreprises PTV, Kaliterre et Opale Conseil.

Je suis pour ma part responsable du module de localisation des sites. Ce module consiste à localiser plusieurs types d'installations : les carrières et sablières d'où sont extraits les principaux matériaux, les plates-formes utilisées pour le stockage, le recyclage ou la transformation de matériaux, et les Installations de Stockage des Déchets Inertes (ISDI) utilisés pour enfouir les matériaux non recyclables et non dangereux. Les produits impliqués sont classés en plusieurs familles : granulats primaires (G), granulats issus du recyclage (g'), enrobés (E), produits issus du recyclage des enrobés (e'), bétons (B), produits non recyclables (NR) et autres produits (O). Le but est de localiser les sites de manière à satisfaire au mieux les demandes émanant d'un ensemble de chantiers. On précise que les demandes peuvent être des livraisons ou des enlèvements de matériaux. La figure 5.2 montre les flux logistiques pris en compte dans le module de localisation.

Deux types de flux de matières cohabitent : les flux directs depuis les carrières et fournisseurs extérieurs vers les chantiers ou les plates-formes puis vers les ISDI, et les flux de retour de matériaux depuis les chantiers vers les plates-formes. Les flux de retour concernent l'enlèvement de produits

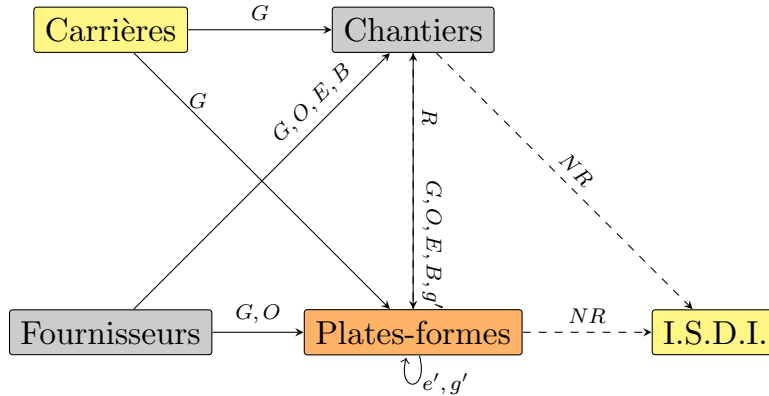


Figure 5.2 – Flux de matières dans le réseau logistique

non recyclables (par exemple de la terre), des matériaux issus de la déconstruction de bâtiments ou des voies de circulation. En particulier, le recyclage des enrobés (RAP, reclaimed asphalt pavement) permet de produire de l'enrobé à partir de matériaux recyclés (jusqu'à un taux de 80%). Le recyclage du béton permet de fabriquer des granulats (g') notamment utilisés en terrassement. Ainsi, une partie du réseau fonctionne en boucle fermée.

Le problème de localisation comporte donc plusieurs échelons, plusieurs produits, plusieurs périodes et est partiellement en boucle fermée. De plus, la production de l'enrobé et du béton nécessite la présence de centrales à enrobé ou à béton sur les plates-formes. Ces centrales sont des ressources critiques pour le système logistique. Nous les modélisons comme des outils mobiles. Des variables binaires associées à la présence de ces centrales sur telle ou telle plate-forme viennent donc enrichir la description du problème. Le problème de localisation est modélisé comme un MILP : la localisation des installations (carrières, plates-formes, ISDI) et des ressources critiques (centrales à béton et à enrobé) est modélisée par des variables binaires ; les flux de matières sont modélisés par des variables continues.

Ce modèle peut être décliné selon trois modes d'utilisation :

- Planification stratégique : sur un horizon de plusieurs années, la demande est simulée sur la base de données macro-économiques. Ce mode d'utilisation permet de planifier les ouvertures ou fermetures de sites ayant une longue durée de vie (carrières et ISDI) et d'aider des décisions de planification foncière.
- Optimisation annuelle du réseau : seules les variables relatives aux plates-formes et aux ressources critiques sont actives, les carrières et ISDI étant figées.
- Réoptimisation locale de la chaîne logistique au démarrage d'un chantier majeur : seuls quelques variables binaires relatives aux plates-formes et aux ressources critiques sont activées, le reste du réseau étant considéré comme figé.

Ce travail est en cours de réalisation. Il a été présenté aux conférences ROADEF 2014 (Péton *et al.*, 2014b) et ISOLDE 2014 (Péton *et al.*, 2014a) et donnera prochainement lieu à une soumission en revue. La phase suivante consistera à introduire dans le modèle une évaluation de l'impact environnemental du réseau logistique, en estimant les émissions de gaz à effet de serre dues aux transports et aux technologies de production. Cela conduira à un modèle bi-objectif qui sera résolu avec la technique de branch-and-bound multi-objectif décrit par Vincent *et al.* (2013).

5.4 Transport urbain mixte passagers/marchandises

Nous avons adapté les travaux sur les problèmes de tournées avec transfert à un problème de distribution de colis en centre ville. Cette problématique de logistique urbaine a fait l'objet de la

thèse d'Anna Trentini (Trentini, 2012), dans le cadre du projet ANR C-GOODS, achevé en 2011. Le but de cette thèse était de concevoir un système mixte de distribution, c'est-à-dire mêlant passagers et marchandises. Dans notre étude, l'idée est d'utiliser la capacité résiduelle d'une ligne de bus pour acheminer des colis depuis un centre de distribution urbain (CDU) vers un certain nombre d'arrêts de bus situés en centre-ville. Concrètement, les colis sont transportés dans des *rolls* qui peuvent être facilement chargés dans les bus au CDU et déchargés aux arrêts de bus. Une fois déchargés, ils sont pris en charge par des tricycles à assistance électrique qui assurent le *dernier kilomètre* de distribution. Un tel fonctionnement est illustré par la Figure 5.3.

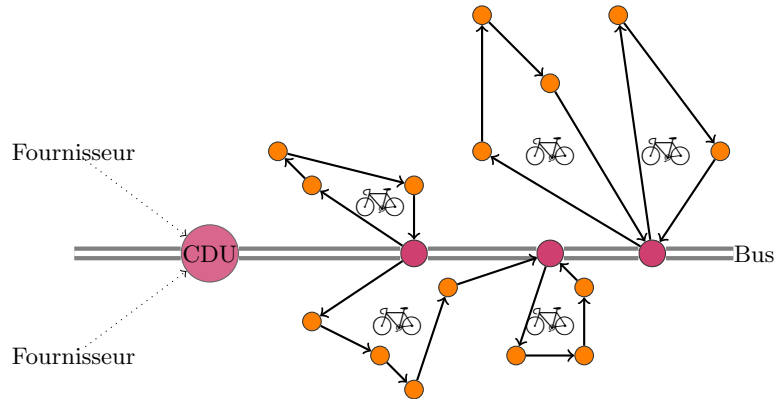


Figure 5.3 – Illustration du système de transport à deux échelons

L'optimisation porte uniquement sur le flux de marchandises, le flux de personnes étant considéré comme une donnée d'entrée. Contrairement à des systèmes plus complexes tels que ceux décrits par Perboli *et al.* (2011), Hemmelmayr *et al.* (2012) ou Crainic *et al.* (2012), un tel système de distribution ne concerne que les flux entrant dans le centre ville et passant par un seul CDU. En revanche, puisqu'il s'appuie essentiellement sur des infrastructures existantes, il ne nécessite qu'une organisation légère et s'applique idéalement aux villes moyennes (Trentini et Malhéné, 2010).

Notre intervention dans ce projet consiste à faire une étude de faisabilité du système mixte de transport de marchandises et de personnes. On cherche à dimensionner la flotte nécessaire pour opérer un tel système. Il est possible de modéliser le problème comme un PDPT avec des contraintes additionnelles, en modélisant les bus comme des véhicules dont la tournée est prédéfinie. Cette approche reviendrait toutefois à résoudre un problème plus général que le problème en question.

Les objectifs à optimiser dans le problème sont le nombre de tricycles utilisés, ainsi que la distance totale parcourue par ces tricycles. Ainsi, l'affectation des rolls aux bus n'a pas d'impact direct sur la valeur de la fonction objectif. Il faut seulement déterminer, pour un ensemble de tournées donné, une affectation des rolls aux bus menant à une solution réalisable. Nous avons donc proposé une décomposition du problème en un problème maître et un problème esclave. Le problème maître consiste à constituer des routes et décider dans quel ordre ces routes sont servies par les tricycles. Cela revient donc à répondre aux questions suivantes :

- Comment la marchandise des clients est-elle répartie dans les rolls ?
- À quel arrêt de bus chaque roll est-il transféré ?
- Dans quel ordre les clients d'une tournée sont-ils visités ?
- Par quel tricycle chaque roll est-il pris en charge ?
- Pour chaque tricycle, quel enchaînement de routes est effectué ?

Le problème esclave (sous-problème de réalisabilité) consiste à affecter les rolls aux bus et à déterminer les heures de service en chaque lieu desservi. Il s'agit donc de vérifier qu'il existe une affectation des rolls aux bus qui respecte les contraintes de capacité des bus et des arrêts, tout en respectant les contraintes horaires du problème.

Le problème maître a été résolu par une variante de l'algorithme ALNS décrit au chapitre 4. Le

problème esclave est résolu au sein de l'heuristique d'insertion de l'ALNS dans un algorithme de *vérification de la réalisabilité des tournées*.

Cette approche a tout d'abord été validée sur un ensemble d'instances aléatoires avec 50 clients, 5 bus et 5 arrêts de bus. Nous avons calculé, en utilisant une méthode de génération de colonnes, des bornes inférieures du nombre de véhicules nécessaires pour satisfaire la demande. Les résultats obtenus par l'ALNS (après 2 minutes de calcul) sont comparés à ces bornes inférieures. Pour 10 des 18 instances, l'ALNS parvient à trouver le nombre minimal de véhicules. Pour les 8 autres instances, l'écart avec la borne inférieure est de 1.

Nous avons ensuite travaillé sur un cas d'étude au centre ville de La Rochelle (Figure 5.4). Le cercle situé au sud représente le CDU, les carrés sont les arrêts de bus.



Figure 5.4 – Zone d'étude au centre ville de La Rochelle

La zone d'étude est traversée par une ligne de bus, nommée *Illico*, qui passe également à proximité d'un CDU situé en bordure sud du centre ville. Au cours de la journée, 74 bus parcourent la ligne. Une enquête de terrain a permis de déterminer les types d'établissement présents dans cette zone et pouvant être intéressés par un service de livraison en tricycles. De même, les taux d'occupation des bus de la ligne *Illico* ont été recensés. Cela a permis de déterminer la capacité résiduelle dans chaque bus.

À partir de ces informations de terrain, nous avons constitué un ensemble d'instances de 105 à 303 clients, représentant divers échantillons de commerces et d'administrations. On associe à chaque client une fenêtre horaire de 1, 2 ou 4 heures de largeur et un temps de service de 5 minutes. Pour chaque sous-ensemble de clients, trois scénarios avec différents niveaux de demande ont été générés.

En résolvant ces instances tests, nous avons pu mettre à jour quelques propriétés intéressantes du systèmes de transport mixte. Ainsi, le nombre de tricycles et de rolls nécessaires dépend bien sûr de la demande, mais surtout de la largeur des fenêtres horaires aux clients. D'autre part, deux arrêts de bus concentrent 60% des flux logistiques, et les deux arrêts situés en bout de ligne ne sont quasiment jamais utilisés.

Étendre ce travail à un réseau avec plusieurs lignes de bus ne pose pas de problème majeur. En revanche, considérer plusieurs CDU ou mélanger des flottes de camions électriques et de tricycles semblent compliquer significativement le problème. Nous avons fait l'hypothèse que tout arrêt

de bus du centre-ville peut être utilisé pour effectuer des opérations de transfert. Sélectionner un sous-ensemble d'arrêts de bus pourrait amener à une solution légèrement moins bonne, mais beaucoup plus simple à mettre en œuvre. Un autre aspect à investiguer concerne la logistique de retour des rolls. Les rolls vides ne deviennent réutilisables qu'une fois retournés au CDU et remplis à nouveau. Une extension souhaitable de notre travail serait donc d'intégrer du retour des rolls dans le modèle, et d'autoriser ainsi plusieurs vagues de préparation de commande dans une même journée.

Cette collaboration a donné lieu à une soumission d'article en juillet 2013 dans EURO Journal on Transportation and Logistics (Masson *et al.*, 2014c) (révision en septembre 2014).

5.5 Temix β

Près de 80% du territoire français concentre moins de 30% de la population. Selon les données de l'INSEE (2009), la croissance démographique de ces zones rurales est supérieure à celle des zones urbaines (+0,9% contre +0,6% par an sur 10 ans). Dans le même temps, l'évolution des modes d'achat, notamment des achats en ligne, permet de se faire livrer partout (ou presque) une quantité de biens toujours grandissante. Le commerce en ligne connaît une progression de 19% par an et a atteint 45 milliards d'euros de chiffre d'affaire en 2012 (source : FEVAD). Mais cette offre présente aussi ses limites et notamment l'accroissement du nombre de véhicules assurant ces livraisons, souvent individualisées.

L'offre de mobilité s'est fortement accrue dans le secteur urbain avec l'arrivée (ou le retour) de tramways, trolleys ou des systèmes de taxis collectifs. La littérature scientifique et technique est très riche sur ces thématiques. Les modèles économiques reposent généralement sur l'existence d'un seuil de rentabilité lié à la densité de la demande. Pour les secteurs ruraux ou péri-urbains, la littérature scientifique abordant cette question n'existe pour ainsi dire pas. La question de la logistique en milieu rural n'est actuellement l'objet d'aucune réflexion, du fait sans doute de l'absence ressentie d'enjeux économiques et de contraintes de fonctionnement.

Le projet TeMix β ¹ entend apporter des réponses sociales, environnementales et économiques à ces phénomènes. Sur le plan économique, ni la livraison des marchandises ni le transport des personnes ne sont viables en milieu rural. Le pari est donc de construire une offre de transport mixte passagers/marchandises permettant de satisfaire la demande au moindre coût.

D'un point de vue recherche opérationnelle, ce projet consiste à produire une estimation des coûts de mise en place et de fonctionnement d'un réseau mixte à l'échelle d'un territoire. Un premier volet consiste à concevoir un réseau de transport de personnes et marchandises, s'appuyant sur une architecture à deux niveaux : des lignes fixes de transport en commun connectés à des lignes de transport à la demande. Ce réseau sera complété par la localisation de sites logistiques de stockage des marchandises. Le second volet est l'optimisation journalière des transports à la demande sur le réseau secondaire et la planification des collectes et livraisons de colis.

Le périmètre de l'étude est le Pays de Redon (30 communes, 67000 habitants), qui possède la particularité de s'étendre sur 2 régions et 3 départements, et de ne pas avoir de service de transport public couvrant l'ensemble du territoire. Le public visé concerne les personnes et la messagerie en B2C ou B2B.

Ce projet est financé par BPI France et les régions Bretagne et Pays de la Loire. Il a été sélectionné suite à l'appel à projet ID4PME du pôle ID4car. Le projet est porté par la société Mobhilis (Ste Marie, 35). Il associe les transports Orain (Messac, 35) et l'École des Mines de Nantes. Il se déroule sur un an, en 2014.

1. URL : <http://www.temix.fr/>

5.6 Projet Végésupply

La logistique collaborative est un levier puissant pour diminuer les coûts et accroître la compétitivité. Le projet Végésupply implique 4 PME horticoles et pépiniéristes situées dans le département du Maine-et-Loire : Pépinières Détriché, Fleuron d'Anjou, Gaignard Fleurs et JCT Plants. Ces entreprises sont positionnées sur une scène concurrentielle à l'échelle internationale. Leur différenciation ne se fait plus uniquement sur la gamme végétale ou le prix mais également sur le service client, en particulier la réactivité et les conditions de livraison.

Ce projet de recherche et développement repose sur la mutualisation de la chaîne logistique de distribution des 4 entreprises. Trois partenaires académiques sont associés : l'École des Mines de Nantes (IRCCyN / SLP), l'École des Mines de Paris (CGS) et l'Université d'Angers (UMR Granem). Le projet a été retenu à l'issue de l'appel FUI 15 et financé par BPI France et la région Pays de la Loire. Il a été labellisé par les pôles de compétitivité Végépolys et Nov@log.

Jusqu'au début 2014, chaque entreprise livrait indépendamment l'ensemble de ses clients sur toute la France. Plus de 100 000 unités d'expédition sont ainsi expédiées chaque année avec deux pics d'activité, un principal en avril-mai et un secondaire à la Toussaint. Ces expéditions de plantes vers plus de 3000 clients français représentent plusieurs millions d'euros de budget annuel. Le transport est effectué par des prestataires logistiques. La principale unité d'expédition est le roll danois (ou roll CC).

Le projet Végésupply consiste à mutualiser l'ensemble de la chaîne logistique de distribution, en construisant un nouveau réseau de distribution (Figure 5.5). Un centre de distribution et de consolidation (CDC) sera créé à proximité des producteurs, et plusieurs centres de distribution régionaux (CDR) sont à localiser sur le territoire français. Le fonctionnement visé consiste en une consolidation possible des flux de produits au CDC, via des navettes locales, puis un transport en camions complets jusqu'aux CDR. Après une rupture de charge au CDR, des tournées de distribution sont alors organisées jusqu'aux clients finaux.

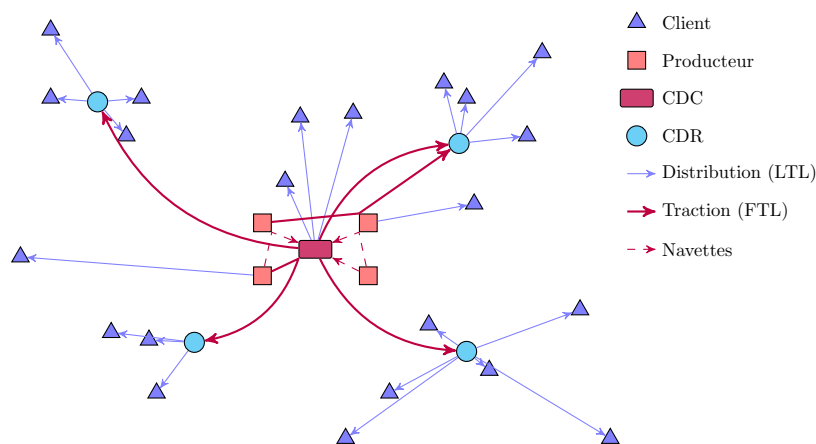


Figure 5.5 – Le réseau logistique mutualisé

Une pré-étude menée en 2012 par l'entreprise 4S Network, pour laquelle nous sommes intervenus en tant que sous-traitants, a permis d'identifier un gain significatif (et confidentiel) lié à cette nouvelle organisation. Suite à cette pré-étude, le projet FUI a été soumis et accepté, pour un démarrage au 1er avril 2013. Notre travail consiste en deux tâches principales : localiser les CDR, optimiser les transports quotidiens sur le réseau ainsi défini. Ces tâches sont effectuées avec mon collègue Fabien Lehuédé et Xin Tang, post-doctorant recruté pour le projet. Rien qu'en France, de nombreux projets de mutualisation logistique ont vu le jour ces dernières années. Pourtant, parmi ceux cités par le rapport du PIPAME (PIPAME, 2011), aucun ne cumule autant de facteurs de difficulté :

1. intégration de décisions stratégiques (localisation des CDR) et opérationnelles (chargement des camions et routage);
2. chaîne logistique multi-niveaux : producteurs, CDC, CDR et clients;
3. structure de coût complexe : le coût d'un camion complet (traction) est fixe, le coût unitaire de distribution des rolls aux clients depuis un CDR décroît lorsque les quantités livrées augmentent ;
4. complexité des flux physiques : si les quantités à transporter et les tarifs le justifient, le passage par le CDC et les CDR n'est pas obligatoire, ce qui engendre une économie financière mais une plus grande complexité de gestion des flux logistiques ;
5. logistique inverse : la recherche d'une politique optimale de retour des rolls dépend du délai de leur remise à disposition par les clients. Or celui-ci est stochastique. Ce volet du projet utilisera des techniques de simulation,
6. répartition des gains entre partenaires : cette répartition a été confiée à l'École des Mines de Paris. Elle utilise la valeur de Shapley (Shapley, 1953).

Nous avons modélisé le problème de localisation des CDR en agrégeant la demande de l'ensemble des clients d'un même département. Ce modèle a été validé par les entreprises et résolu à l'aide de Cplex 12.5. Les travaux actuels consistent à accélérer les temps de calcul par le recours à des heuristiques telles que celles décrites au chapitre 2, et de tester l'impact d'inégalités valides ou des coupes heuristiques (pouvant avec une probabilité très faible couper des solutions optimales). Ce travail sera ensuite porté sur un solveur libre en vue de sa mise en production. Un article sera soumis à l'automne 2014 (International Journal of Production Economics, ou revue similaire)

Le problème de routage des produits dans le réseau de distribution est un MILP impliquant plusieurs décisions successives : (i) savoir à quel CDR est rattaché chaque client ; (ii) comptabiliser le nombre de camions livrant chaque jour les CDR depuis la région de production en précisant leur point de départ exact (CDC ou producteur) ; (iii) lister les navettes qui approvisionnent les CDC depuis les producteurs.

Nous avons travaillé sur un modèle légèrement simplifié, dont les variables ont été décomposées selon les trois décisions successives présentées ci-dessus. Ce travail préliminaire a permis d'accélérer la résolution de manière significative tout en garantissant un écart d'au plus 1.7% par rapport à une borne inférieure fournie par Cplex. Ce travail a été présenté à la conférence ROADEF 2014 (Tang *et al.*, 2014). Nous prévoyons de rédiger un article pour le soumettre avant le 31 octobre 2014 à l'édition spéciale de OR Spectrum intitulée *Collaborative Planning in Transportation*.

5.7 Thèse 4S Network

L'entreprise 4S Network² a pour mission d'accompagner ses clients dans la réflexion, la conception et l'organisation de solutions Supply Chain novatrices et durables afin de répondre efficacement aux défis de demain. Elle est l'instigatrice des *Centres de Routage Collaboratifs* (CRC). Le principe est de concentrer les flux de plusieurs industriels au niveau régional sur un point de livraison unique, le CRC, pour optimiser les chargements. Les industriels ont ainsi les moyens de grouper leur chargement à destination de plusieurs clients. Et les clients peuvent être livrés en camions complets en provenance de plusieurs industriels. Le premier CRC est actuellement en phase de lancement.

Une des conditions de succès des CRC est qu'ils puissent attirer les flux d'un nombre suffisant d'entreprises partenaires. À cet effet, nous avons été chargés de développer un simulateur qui évalue les gains financiers et environnementaux qu'apporte l'utilisation des CRC pour une entreprise de production, un distributeur et/ou un prestataire logistique voulant apporter une solution nationale

2. <http://4snetwork.com>

à ses clients. Ce travail s'inscrit dans le cadre de la thèse de doctorat de Juliette Médina (2013–2016), co-encadrée avec Fabien Lehuédé. Il nécessitera de résoudre des problèmes d'optimisation à variables mixtes généralisant ceux abordés dans le projet Végésupply.

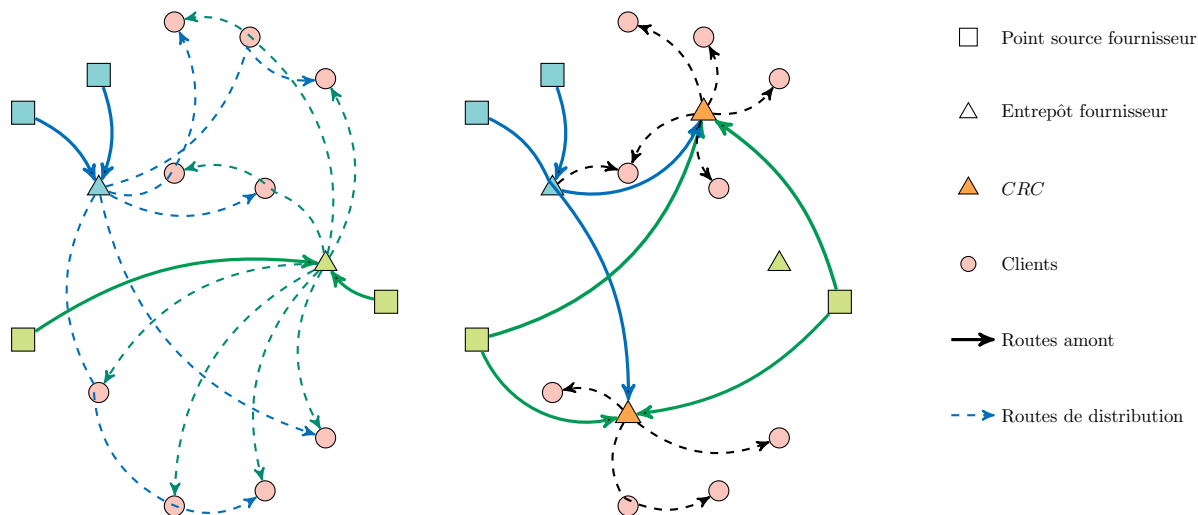


Figure 5.6 – Réseau de distribution sans CRC (à gauche) et avec CRC (à droite)

L'apport des CRC est illustré par la Figure 5.6. La partie gauche montre un réseau logistique classique où chaque entreprise route ses flux via ses propres entrepôts. Il met en jeu plusieurs types d'acteurs : des fournisseurs ou *sources* qui disposent des unités logistiques (produits formant les commandes clients), des *clients* qui passent commandes des unités logistiques et des *entrepôts fournisseurs* propriété de chaque entreprise.

La partie droite montre le rôle des CRC. Les fournisseurs envoient leurs unités logistiques vers l'un des CRC les plus proches. Par la suite, les commandes sont regroupées par client dans les CRC et distribuées via les routes de distribution. Le gain kilométrique est possible grâce à la multiplicité des CRC qui permet d'empêcher des détours non souhaités. En présence de coûts marginaux décroissants, un gain financier est en outre assuré par le co-chargement de camions aux CRC. Une dernière hypothèse est que des commandes peuvent être séparées en plusieurs lots, qui sont regroupés au CRC, afin d'optimiser le chargement dans les camions. À terme, l'utilisation d'un tel réseau collaboratif vise à rendre économiquement viable l'expédition de commandes de faible volume à haute fréquence.

Chapitre 6

Conclusion générale

6.1 Choisir de ne pas choisir...

Ce mémoire porte sur deux thématiques principales : d'une part les problèmes de localisation et de design de chaînes logistiques et d'autre part les problèmes de tournées de véhicules pour l'optimisation du transport de personnes. J'ai travaillé sur ces deux thématiques de manière complètement disjointe pendant plusieurs années. Cette situation m'a conduit à me demander s'il fallait privilégier l'une ou l'autre thématique, afin de donner davantage de cohérence à mes travaux.

Il est en effet difficile de maîtriser l'état de l'art sur deux thématiques, mêmes proches. Professionnellement parlant, il est sans doute plus sage d'attendre d'être reconnu dans un domaine avant de lancer des travaux dans un deuxième domaine, même très proche. En revanche, travailler sur un spectre large d'applications favorise le décroisement et permet d'être pertinent sur un plus grand nombre de projets.

Ma culture des problèmes d'optimisation des transports remonte à ma formation initiale à l'Institut de Mathématiques Appliquées (Angers). Ces problèmes regroupent une communauté nationale active, dans laquelle il aurait été facile de s'inscrire complètement. Au fil des projets de recherche et des conférences, je me suis davantage retrouvé impliqué dans la communauté scientifique sur les problèmes de localisation, notamment à travers l'Euro Working Group on Locational Analysis (EWGLA).

À force de ne pas choisir entre ces deux domaines, j'ai fini par travailler sur des projets, comme le projet Végésupply, comportant à la fois un volet localisation et un volet tournées de véhicules. Sans encore former un corpus cohérent, mes activités de recherche, initialement dispersées, ont donc fini par former un ensemble connexe, comme le montre le chapitre précédent. Cette connexion entre les différents travaux s'est matérialisée par l'utilisation récurrente de techniques de résolution comme la recherche à voisinage large, utilisé abondamment pour résoudre des problèmes de tournées de véhicules, et aujourd'hui des problèmes de localisation. Elle s'est également établie grâce à des notions transversales comme les pratiques de mutualisation, aussi bien valables pour du transports de personnes que pour la distribution de biens dans une chaîne logistique.

6.2 Perspectives futures

Depuis ma formation initiale jusqu'à la rédaction de mon mémoire d'habilitation à diriger des recherches, j'ai pu mesurer à deux reprises à quel point il est difficile d'exercer une réorientation thématique. À l'issue de ma formation initiale, j'avais acquis une formation essentiellement axée

sur l'optimisation combinatoire. J'ai ensuite effectué mon doctorat dans le domaine des méthodes de points intérieurs pour l'optimisation convexe non différentiable, avant de revenir en quelque sorte à mes premières amours à l'occasion de mon recrutement à l'École des Mines de Nantes.

Ce parcours m'a donné une vision très large de la recherche opérationnelle, chose que j'apprécie aujourd'hui. Mais pendant de longues années, il m'a été difficile de capitaliser sur des travaux précédents. Ouvrir de nouvelles thématiques est certes intellectuellement très motivant, mais professionnellement peu productif. Depuis quelques années, j'ai enfin pu accumuler un certain nombre de travaux sur des thématiques voisines. Bien que naturellement curieux d'autres thématiques, je compte poursuivre plusieurs années encore mon activité dans le domaines des modèles et méthodes d'optimisation combinatoire pour résoudre des problèmes de logistique et de transport. Les perspectives de recherche évoquées en conclusion des chapitres 2 à 4 montrent qu'il y a de la matière pour cela.

Une manière idéale de concilier localisation et transports serait de travailler sur le problème conjoint de localisation-routage. Ce problème est étudié depuis plusieurs dizaines d'années, mais l'état de l'art récent de Prodhon et Prins (2014) montre un regain d'intérêt récent, probablement lié à l'émergence de pratiques telles que les systèmes de distribution multi-niveaux, la logistique collaborative, la distribution multi-canal ou la logistique urbaine. Il apparaît plus évident que jamais pour les décideurs que les questions de localisation des sites logistiques et d'optimisation des transports sont interdépendantes. Malgré tout, ces décisions se prennent généralement de manière asynchrone et par des personnes différentes. Les approches de décomposition des décisions dans des systèmes logistiques complexes ont donc encore de beaux jours à vivre.

L'accélération des cycles industriels et l'apparition de nouvelles pratiques de distribution impliquent des reconfigurations de plus en plus fréquentes des réseaux logistiques. Plus on s'approche du client final, plus le réseau devient agile. Si on y ajoute des contraintes métier et des considérations liées à la qualité de service ou à l'impact environnemental, cela engendre des modèles d'optimisation riches probablement très difficiles à résoudre de manière optimale.

Ces dernières années ont vu un développement rapide des solutions de mobilité, de traçabilité ou de remontée d'information du terrain en temps quasi-réel. Avec la démocratisation des outils cartographiques, le développement de solveurs libres accessibles à des TPE et l'ouverture des données publiques, connecter les algorithmes de recherche opérationnelle à l'utilisateur devient de plus en plus facile. Ainsi de nombreux phénomènes, qui apparaissent au citoyen comme une complexification du monde, peuvent être considérés comme des opportunités à saisir pour la recherche opérationnelle.

6.3 Un métier en évolution

À l'école des Mines de Nantes, j'ai la chance de pouvoir exercer mes missions d'enseignement, de recherche et de valorisation dans un champ thématique cohérent : production, logistique, transport, recherche opérationnelle. En revanche, mes tutelles académiques connaissent actuellement de profondes mutations. Le Département Automatique-Productique, dans lequel je travaille, est en train de se rapprocher du Département Informatique. Deux ans après la création de l'Institut Mines Télécom, l'École des Mines de Nantes a annoncé en mai 2014 un projet de fusion avec Télécom Bretagne. Plus globalement, L'UNAM et l'UEB sont en train de constituer une communauté d'universités et établissements (COMUE) appelée Université Bretagne Loire. Pour finir, l'IRCCyN est en cours de rapprochement avec le Laboratoire d'Informatique de Nantes (LINA). On peut voir ces bouleversements comme une menace car ils viennent bousculer des habitudes bien établies. On peut aussi les voir comme une opportunité de mieux structurer le paysage local et régional de la recherche. Très occupé par la réalisation des projets de recherche et de valorisation, je n'ai pas souhaité participer à ces diverses réflexions, qui auront pourtant un impact important dans les années à venir.

Le mode de pilotage de la recherche scientifique a également beaucoup évolué, en basculant vers un mode de financement par projets (Investissement d'Avenir, Projets européens, ANR, projets régionaux etc.). Il me semble que ce mode de financement favorise les projets de recherche opérationnelle, qui peuvent être fondamentaux ou appliqués, purement académiques ou en partenariat avec l'industrie, de grande envergure ou très modestes. Notre discipline peut donc prétendre répondre à une large gamme d'appels à projet. Plusieurs risques existent toutefois pour les chercheurs. Tout d'abord celui d'une recherche au coup par coup, se positionnant de manière opportuniste sur certains appels à projets, sans garantie de garder un fil conducteur. Ensuite, la plus grande difficulté à maîtriser sa charge de travail dans le temps, et à gérer des équipes à géométrie variable au gré des recrutements temporaires de collaborateurs. L'expérience acquise ces dernières années me permet aujourd'hui de mieux "sentir" les projets sur lesquels me positionner, afin de concilier l'intérêt scientifique des projets, la cohérence de mes thématiques de recherche et la planification dans le temps de ma charge de travail.

La création des pôles de compétitivité et les pratiques de labellisation mises en place par ces pôles obligent les chercheurs à rester au contact des réalités économiques. Comme il n'existe pas de pôle de compétitivité dans mes domaines de compétence en région Pays de la Loire, il est plus que jamais nécessaire de s'appuyer sur des pôles extérieurs tels qu'ID4car ou Nov@log et de développer des réseaux nationaux et internationaux. Côté univers aussi variés que des établissements médico-sociaux, des entreprises horticoles ou de travaux publics m'a permis d'élargir ma vision des réalités socio-économiques. À titre personnel, je suis attiré par plusieurs autres domaines d'application tels que la logistique urbaine, l'impact des nouvelles technologies de communication sur les modes de consommation et de déplacement, et dans un tout autre registre l'optimisation logistique de filières telles que celle de la biomasse ou des énergies nouvelles.

Quelle que soit ma future orientation de carrière, je souhaite poursuivre l'alternance entre projets de recherche académique et de recherche appliquée, et que mes diverses missions puissent continuer à s'enrichir mutuellement.

Chapitre 7

Curriculum Vitae

7.1 Etat Civil et coordonnées

Nom : Péton
Prénom : Olivier
Date de naissance : 28 avril 1972
Lieu de Naissance : Landerneau (29)
Nationalité : Française

Poste actuel :

Maître Assistant à l'Ecole des Mines de Nantes (EMN)
Département Automatique Productique
4 rue Alfred Kastler, 44307 Nantes Cedex 3
Téléphone : +33 2 51 85 83 13
Mél : olivier.peton@mines-nantes.fr

Laboratoire de Rattachement :

Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN),
UMR CNRS 6597
1, rue de la Noë, 44321 Nantes
Équipe de Rattachement : Systèmes Logistiques et de Production (SLP)

7.2 Formation et parcours professionnel

Parcours professionnel

depuis 08/2005 : Maître Assistant, École des Mines de Nantes
08/2004 – 07/2005 : Maître Assistant Associé, École des Mines de Nantes
2003-2004 : Attaché Temporaire d'Enseignement et de Recherche (mi-temps), Université de Versailles Saint-Quentin-en-Yvelines (UVSQ), UFR des Sciences
2003 : Qualification aux fonctions de Maître de Conférences, section 26 et 27.
1996-2002 : Assistant, Université de Genève, Faculté des Sciences Économiques et Sociales.

Formation

- 2002 : Doctorat en Sciences Économiques et Sociales, de l'Université de Genève
 Titre : “*The homogeneous analytic center cutting plane method*”
 Soutenu le 5 juillet 2002 devant le jury composé de
 Prof. Jean-Philippe VIAL, Université de Genève, directeur de thèse
 Prof. Yurii NESTEROV, Université catholique de Louvain, rapporteur
 Prof. Frédéric BONNANS, Université Paris Sorbonne et INRIA, rapporteur
 Prof. Manfred GILLI, Université de Genève, président du jury
 Prof. Alain HAURIE, Université de Genève, examinateur
- 1999 : Cours post-grade en Statistiques et Recherche Opérationnelle, EPFL Lausanne et Université Joseph Fourier, Grenoble.
- 1996 : DEA en Informatique et Recherche Opérationnelle, Université Pierre et Marie Curie, Paris 6, mention Très bien.
- 1994 : DESS en Mathématiques Appliquées et Informatique, Université Catholique de l'Ouest, mention Bien.
- 1994 : Diplôme de l'Institut de Mathématiques Appliquées, Université Catholique de l'Ouest, Angers.

7.3 Activités d'encadrement

7.3.1 Encadrements de doctorat

1. Thèse de Phuong Nga Thanh :
Conception et planification stratégique des réseaux logistiques complexes. Bourse École des Mines de Nantes (2005–2008), co-encadrement (50%) avec Nathalie Bostel, direction : Christelle Guéret-Jussien. Soutenance le 23 octobre 2008.
2. Thèse de Valentina Gutiérrez :
Combined routing and staff scheduling in healthcare logistics. Bourse Universidad de Los Andes (2008–2010), co-encadrement (40%) avec Christelle Guéret-Jussien, Ciro Amaya et Nubia Velasco, thèse arrêtée pour raisons médicales.
3. Thèse de Renaud Masson :
Problèmes de collectes et livraisons avec transferts. Bourse MENRT (2009–2012), co-encadrement (30%) avec Fabien Lehuédé, direction : Nathalie Bostel. Soutenance le 28 septembre 2012.
4. Thèse de Majid Eskandarpour :
A large neighborhood search for sustainable supply chain network design. Bourse région Pays de la Loire – projet OLASI, (2011–2014) co-encadrement (40%) avec Pierre Dejax et Joe Miemczyk. Soutenance prévue en novembre ou décembre 2014.
5. Thèse de Juliette Médina :
 Titre provisoire : *Modèles et méthodes d'optimisation pour la mutualisation des chaînes logistiques*. Demi-bourse EMN et contrat 4S Network (2013–2016), co-encadrement (30%, puis 50% après HDR) avec Fabien Lehuédé, direction : Nathalie Bostel.

7.3.2 Autres encadrements

1. Thomas Vincent. Post-doctorant sur le projet ORLoGES : *Optimisation des opérations logistiques d'une entreprise de travaux publics* (2014, 12 mois).
2. Minh Hoàng Hà. Post-doctorant sur le projet ORLoGES : *Conception de réseau logistique d'une entreprise de travaux publics* (2013, 12 mois).
3. Xin Tang. Post-doctorat sur le projet Végésupply. *Localisation et optimisation des transports dans un réseau de distribution mutualisé* (2013-2015, 24 mois).

4. Olivier Quirion-Blais, doctorant Polytechnique Montréal. *Résolution de problèmes d'arc routing appliqué au déneigement des routes*, visiteur de mars à juin 2014.
5. Nicolas Bioret. Ingénieur de recherche *Développeur Javascript* sur le projet Temix β (2014, 6 mois).
6. Guy-Armand Bado. Stage de master (Université de Poitiers), Projet TeMix. *Conception et optimisation d'un réseau de transport mixte passagers-marchandises* (2014, 6 mois).
7. Arthur Brunel. *Conception d'un réseau de distribution partagé*. Stage Master ORO (2012, 6 mois).
8. Agnès Le Roux, *Planification de l'activité du personnel soignant pour l'hospitalisation à domicile*. Stage Master (2011, 6 mois).
9. Irene Cosentino. Stage de master (Université de Catane), projet Marika (2008, 3 mois).
10. Aurélie Robert. *Conception dynamique de réseaux logistiques complexes : un modèle multi-niveaux intégrant les fournisseurs et le concept de nomenclature*. IMA, Angers (2005) 4 mois.

7.3.3 Jury de thèses de doctorat

1. Ahmed Atahran. *Étude et résolution d'un problème de transport à la demande multicritère*. Université François Rabelais, Tours, décembre 2012 (Examineur).
2. Anna Trentini. *Proposition d'un système de transport urbain mixte : application dans le cadre de la ville moyenne de La Rochelle*. EIGSI La Rochelle et Mines ParisTech, La Rochelle, décembre 2012 (Membre invité).
3. Christine Vanovermeire. *Horizontal collaboration in logistics : increasing efficiency through flexibility, gain sharing and collaborative planning*. University of Antwerp, janvier 2014 (Examineur).
4. Juan Carlos Rivera. *Thématique : problème de tournées de véhicules, logistique humanitaire*. Université de Technologie de Troyes, octobre 2014 (Examineur).

7.3.4 Comités de suivi de thèse

1. Minh Hoàng Hà (2009-2012). *Modélisation et résolution de problèmes généralisés de tournées de véhicules*. École des Mines de Nantes et École Polytechnique de Montréal.
2. Philippe Grangier (2012-2015) *Problèmes de synchronisation dans les tournées multi-échelons ou multi-modales*. École des Mines de Nantes et École Polytechnique de Montréal.

7.3.5 Jury de thèses de master

1. Fouad Maach, *Bi-objective multi-assignment capacitated location-allocation problem* Master of Science, Virginia Institute of Technology (2007).
2. Diana C. Flórez, *A mathematical programming approach to airline crew pairing optimization*, Maestria, Universidad de Los Andes, Bogota (2009).

7.4 Implication dans la communauté scientifique

7.4.1 Organisation de congrès scientifiques

- Responsable de l'organisation du congrès EWGLA2011 (Euro Working Group in Locational Analysis) à Nantes, du 12 au 14 octobre 2011 (90 participants).
- membre du comité d'organisation du Network Optimization Workshop (NOW 2015) à La Rochelle, du 18 au 20 mai 2015.

- Membre du comité d'organisation du congrès VeRoLog 2016 à Nantes en juin 2016.
- Membre du comité scientifique des congrès ROADEF2006 (Lille), CIE39 (Troyes, 2009), ICM-SAO2009 (Sharjah), EWGLA2010 (Naples), ROADEF2012 (Angers), ROADEF2013 (Troyes), ISOLDE2014 (Naples/Capri), ROADEF2015 (Marseille), Odysseus 2015 (Ajaccio), EWGLA 2015 (Budapest).
- Organisation de sessions lors des congrès EURO2015 (Glasgow), ROADEF 2010 (Toulouse) et 2011 (St Etienne).
- Arbitrage de soumissions pour les congrès ICMSAO09, CIE39, MIC2011 (Udine), EWGLA2011, ROADEF2012, ROADEF2013, ROADEF2015, ISOLDE2014.

7.4.2 Rapporteur pour des revues internationales

- Annals of Operations Research
- Asian Pacific Marketing Research
- Computers and Industrial Engineering
- Computers and Operations Research
- European Journal of Operational Research
- International Journal of Engineering Management and Economics
- International Journal of Operations and Quantitative Management
- International Journal of Production Economics
- Networks
- Optimization Letters
- Scientific Research and Essays
- Transportation Research part E

7.4.3 Rapporteur de projets

- Rapporteur pour un projet ANR en 2013
- Rapporteur pour un projet de Postdoctoral Fellowship (FWO - Research foundation Flanders) 2014

7.4.4 Animation de la recherche

- EURO Working Group on Locational Analysis (EWGLA) : membre élu du coordination board pour la période 2015-2018, avec Jörg Kalcsics (Karlsruhe Institute of Technology) et Sergio García Quiles (University of Edimburgh)
- Organisation d'un séminaire d'équipe SLP sur 3 jours en mai 2008 (La Pommeraye, 49)
- Organisation de 2 séminaires d'équipe SLP sur une journée, en janvier 2012 et juin 2013
- Organisation de 35 séminaires scientifiques (chercheurs invités, collègues, doctorants) depuis 2009

7.4.5 Affiliation à des sociétés savantes et groupes de travail

- EURO : Association of European Operational Research Societies
- EU/ME : EURO working group, European chapter on metaheuristics
- EWGLA : Euro Working Group in Locational Analysis
- Euro Working Group on Sustainable Supply Chains
- VeRoLog : EURO Working Group, Vehicle Routing and Logistics Optimization
- ROADeF : Société française de recherche opérationnelle et d'aide à la décision
- GT2L : Groupe de travail du GdR Recherche Opérationnelle du CNRS sur le transport et la logistique
- GDR MACS, groupes FL et META.

7.4.6 Séjours académiques

- Université de Sharjah (Emirat Arabes Unis), avril 2008, 1 semaine, Nadjib Brahim
- Université de Los Andes (Bogota, Colombie), avril 2009, 2 semaines, encadrement thèse Valentina Gutiérrez
- Université de Vienne (Autriche), 2×1 semaine en juin 2009 et 2010 : DARP multi-objectif.
- Université de Lisbonne, mars 2012, 1 semaine, problèmes de localisation, Francisco Saldanha da Gama
- CIRRELT, Montréal, mai 2014, 1 semaine.

7.5 Subventions de recherche et contrats

1. **Projet Marika**

- Objectif : Conception d'un logiciel de tournées de véhicules pour optimiser le transport de personnes handicapées mentales vers des établissements médico-sociaux.
- Période : 2007–2008
- Financement : ADAPEI 44, 39k€

2. **Projet cTouVeR : Conception de Tournées de Véhicules Régulières**

- Objectif : Confection de plans de transport conciliant la minimisation des coûts et la régularité des horaires de passage chez chaque client, pour un problème de tournées multi-périodiques avec une demande irrégulière.
- Période : 2008–2009
- Partenaires : Dominique Feillet (EMSE), Thierry Garaix (Politecnico Torino), Fabien Lehuédé (EMN), Dominique Quadri (Univ. Avignon)
- Financement : GdR RO, 4.5k€

3. **Projet MC-DARP : multicriteria dial-a-ride problem**

- Objectif : Résolution d'un problème de type DARP optimisant plusieurs objectifs liés à la performance logistique et à la qualité de service.
- Période : 2009–2010
- Partenaires : Fabien Lehuédé et Renaud Masson (EMN), Sophie Parragh et Fabien Tricoire (Univ. Vienne, Autriche)
- Financement : EGIDE, projet Hubert Curien Amadeus, 4.2k€

4. **Projet Végésupply**

- Objectif : Développer une organisation novatrice, mutualisée et durable pour accroître la compétitivité des entreprises du végétal, soumises à de nombreuses contraintes en matière de transport.
- Période : 2013–2016
- Partenaires : Fleuron d'Anjou, Gaignard Fleurs, JCT PLant, Pépinières Détriché, GRANEM (Univ. Angers), École des Mines de Paris
- Financement : FUI 15, Financement BPI France de 154k€

5. **Thèse Juliette Médina**

- Titre : Modèles et méthodes d'optimisation pour des pratiques de logistique collaborative
- Objectif : Conception d'un outil innovant d'aide à la décision pour des entreprises s'engageant dans une démarche de logistique collaborative
- Période : 2013–2016
- Partenaires : 4S Network
- Financement : Ecole des Mines de Nantes, Financement 4S Network de 117 k€

7.6 Participation à des projets de recherche

1. **Projet Age d'Or Services**

- Objectif : Assistance à la mise en place d'un logiciel d'aide à la décision pour la planification des activités de services à la personne
 - Période : 2010
 - Participation : étude de marché sur l'offre logicielle
 - Financement : Age d'Or services, 12 k€
- 2. Projet MeetMe**
- Objectif : optimisation de l'organisation de bourses d'échanges de technologies
 - Période : 2008
 - Participation : participation à la modélisation du problème de planification
 - Partenaires : Capsulae
- 3. Projet OLASI**
- Titre : Optimisation des réseaux Logistiques, stratégie Achats et Systèmes d'Information : entreprise étendue et performance
 - Objectif : fédérer et coordonner les équipes de recherche régionales autour des problématiques de la chaîne logistique globale et des achats. Développer des modèles novateurs d'organisation et d'optimisation de la chaîne logistique.
 - Période : 2010-2014
 - Participation : encadrement de la thèse de Majid Eskandarpour
 - Partenaires : Audencia, Cret-Log, Université de Nantes
 - Financement : Projet d'émergence collective, Région Pays de la Loire. Subvention de 500k€ pour l'ensemble du projet.
- 4. Projet Fusion CO2 : Modélisation de la performance durable de la supply chain et des relations fournisseurs**
- Objectif : Fournir des éléments conceptuels et empiriques sur les mécanismes de formation, en amont de la supply chain, de la performance durable et écologique des entreprises, avec plus spécifiquement une application à l'organisation des transports.
 - Période : 2010-2012
 - Participation : encadrement du stage d'Arthur Brunel, état de l'art
 - Partenaires : Audencia, CretLog, DISP Lyon
 - Financement : Prédit/ ADEME
- 5. Pré-Etude Végémarché**
- Objectif : Estimation des coûts d'un système de distribution mutualisée
 - Période : 2012
 - Participation : modélisation du problème
 - Partenaires : sous-traitance pour 4S Network
 - Financement : 8k€
- 6. ORLoGES : Optimisation du Réseau Logistique en Génie civil, avec les aspects Économiques et Sociétaux**
- Objectif : Réalisation d'une suite logicielle pour l'optimisation des opérations logistiques d'une entreprise de travaux publics.
 - Période : 2012-2015
 - Participation : Responsabilité du lot 4 : localisation
 - Partenaires : Groupe Luc Durand, PTV, Kaliterre, Opale Conseil
 - Financement : Investissements d'Avenir. 253k€
- 7. Projet TeMix**
- Titre : Transport mixte personnes+merchandises dans les territoires de faible densité
 - Objectif : Concevoir une offre de transport à la demande mixant transport de personnes et de marchandises pour des territoires ruraux ou péri-urbains.
 - Période : 2014
 - Partenaires : Mobhilis, Transports Orain
 - Financement : 32 k€ (BPI, Régions Bretagne et Pays de la Loire. Appel à projets ID4PME)

7.7 Activités d'enseignement

Cette section détaille principalement mes activités d'enseignement à l'EMN depuis 2004. Mes expériences précédentes sont synthétisées dans le tableau 7.1.

Lieu	Années	Niveau	Contenu
IMA Angers	1991-92	L1	Monitorat, TD d'Analyse (60h)
Uni. Genève	1996-2002	L3 à M2	Poste d'assistant. Plusieurs centaines d'heures d'enseignement et encadrement : gestion de production, prévisions, gestion des stocks, files d'attente, stages et projets étudiants
UVSQ	2003-04	M1, M2	recherche opérationnelle, optimisation continue, algorithmique (100h)
ESC Rennes	2003-04	M1, M2	facility location, facility layout, prévisions, programmation linéaire, graphes. 75h + corrections de projets et études de cas
EMN (vacation)	2003-04	M2	programmation mathématique (30h)

Tableau 7.1 – Synthèse de mes expériences d'enseignement de 1991 à 2004

7.7.1 Cours enseignés

À l'EMN, les charges d'enseignement sont évaluées en *Unités Pédagogiques* (UP). Le nombre d'UP attribuées pour un enseignement dépend de la forme pédagogique (cours, TP, encadrement), du nombre d'étudiants concernés (petites classes, grandes classes). Toutes les activités d'enseignement, d'encadrement (projets tutorés et stages) ainsi que les responsabilités pédagogiques donnent lieu à une comptabilisation des UP. Les enseignements sont regroupés en *Unités de Valeur* (UV), d'une durée variant entre 45h et 75h. Pour se fixer les idées, une heure de TP ou TD compte pour 1 UP, un encadrement de projet de fin d'études avec visite compte pour 12 UP, la responsabilité d'une UV compte de 10 à 20 UP.

La quasi-totalité de mes charges d'enseignement a lieu au département automatique-productique. J'interviens en années 2 et 3 de la formation ingénieur (niveau M1 et M2), dans les options GOPL (Gestion des Opérations en Production et Logistique) et QSF (Qualité et Sûreté de Fonctionnement). J'interviens également aux niveaux M1 et M2 du master international. Celui-ci s'est appelé MLPS (Management of Logistics and Production Systems) jusqu'en 2011 et MOST (Management and Optimization of Supply chains and Transport) depuis 2012.

Mes enseignements peuvent être partagés en 2 grands domaines : la recherche opérationnelle et le génie industriel. Le tableau 7.2 récapitule mes charges d'enseignement en recherche opérationnelle. Le tableau 7.3 récapitule mes charges d'enseignement en génie industriel. Globalement les cours ayant un intitulé en français concernent la formation ingénieur, les autres concernent les masters. Les variations du nombre d'UP selon les années s'explique par les variations des volumes horaires d'enseignement mais aussi par des modifications des règles de calcul des UP.

L'encadrement de projets représente une part non négligeable de mes activités d'enseignement. Ces projets sont listés dans le Tableau 7.4. Le nombre d'heures indiqué est celui figurant à l'emploi du temps pour les élèves. Le temps effectif d'encadrement est très variable d'un projet à l'autre.

Le Tableau 7.5 liste mes encadrements de projet de fin d'étude à l'EMN. Ces stages a une durée de 6 mois.

Cours	Années	Niveau	Langue	UP
Programmation Mathématique	2004-05	M1	Fr.	7.5
	2005-06			28.1
Operations Research	2010-11	M1	Eng.	27
	2011-12			27
	2012-13			32.4
	2013-14			30
	2013-14	M2	15	
Optimisation	2008-09	M2	Fr.	9
	2009-10			16.5
	2010-11			16.5
	2011-12			16.5
	2012-13			18
	2013-14			33
Modèles stochastiques et systèmes de production	2004-05	M2	Fr.	15
	2005-06			9

Tableau 7.2 – Tableau récapitulatif des enseignements en recherche opérationnelle

Cours	Années	Niveau	Langue	UP
Gestion de la Production	2004-05	M1	Fr.	40
	2005-06		Fr.	24
	2006-07		Eng.	51
	2007-08		Eng.	51
	2008-09		Fr.	30
	2009-10		Fr.	24
	2010-11		Fr.	33.5
	2011-12		Fr.	33.75
	2012-13		Fr.	33.75
	2013-14		Fr.	33.75
Logistique d'entreposage	2010 à 2012	M1	Fr.	5
Production Management and Production Planning	2004-05	M1	Eng.	49.5
	2005-06			38.3
	2007-08			12
	2008-09			12
	2009-10			36
	2010-11			35
	2012-13			36
2013-14	24			
Etudes de Cas (en logistique)	2005 à 2012	M2	Fr.	40
Visites, salons etc.	2004 à 2009	M2	Fr. Eng.	34

Tableau 7.3 – Tableau Récapitulatif des enseignements en génie industriel

Année	Public	Nombre	Type de projet	Heures	Entreprises/ Thématique
2004-05	M1	0.5	Projet OSE	60h	Open Shop
2005-2006	M1	1	Projet OSE	60h	PileJe / Prévisions de vente
	M1	1	Rechercher & Innover	10h	Bolivia Inti / four solaires
	M2	1	Projet d'option	90h	Alcan / Ordonnancement
2006-2007	M1	2	Projet OSE	60h	Bolivia Inti / four solaire
	M2	0.5	Projet d'option	90h	Système U / distribution
2007-2008	M1	0.5	Projet OSE	60h	Cartographie
	M2	1	Projet d'option	90h	Pronuptia / Affectation
	M2	1	LPS Project	90h	Category management
2008-2009	M2	0.33	Projet d'option	90h	Event scheduling
	M2	1	LPS Project	90h	Category management
2009-2010	M1	1	Rechercher & Innover	10h	Transport à la demande
2010-2011	M2	0.5	Projet d'option	90h	But / Distribution
	M2	1	LPS Project	90h	Transport, affectation
2012-2013	M2	0.5	Projet d'option	90h	Distribution mutualisée
2013-2014	M2	1	SCIProject	60h	CMN/ statistiques
	M2	0.5	SCIProject	60h	Sol3D / Biomasse, transport
	M2	0.5	Projet d'option	60h	Sol3D / Biomasse, localisation

Tableau 7.4 – Encadrement de projets tutorés

Année	Public	Nombre	UP	Entreprises
2004-05	GOPL	3.5	42	Air Liquide, Daher, Transpac, Vuitton
	MLPS	1	12	Michelin
2005-2006	GOPL + M2 ASP	1	12	Total
	GOPL	2	36	Rolex, Scheider Electric
2006-2007	GOPL + MsC VT	1	12	France Telecom
	GOPL	4	48	BNPParibas, Elidis, Keolis, Valeo
2007-2008	MLPS	1	12	Auchan
2007-2008	GOPL	2	24	Danone, Université Saragosse
2008-2009	GOPL	2	24	Coca Cola, Système U
	MLPS	1	12	Reckitt Benckiser
2009-2010	GOPL	3	36	Danone, Labinal, Thales Alenia Space,
	MLPS	1	12	Airbus
2010-2011	GOPL	2	24	L'Oréal, Orange
2011-2012	GOPL	3	36	But, Louis Vuitton (× 2)
	MLPS	1	12	Technicolor
2012-2013	GOPL	3	28	Cecab, L'Oréal, Schneider Electric
2013-2014	GOPL	2	28	Alcon, Armor
	MOST	2	28	fastercom.ca, Lafarge

Tableau 7.5 – Encadrement de projets de fin d'étude (niveau M2)

Le Tableau 7.6 synthétise mes heures d'enseignement par année depuis 2004. Une charge normale est de 180 UP par an.

Année	Responsabilités Pédagogiques	Cours	Encadrement et jurys	Total
2004-05	84	146	84	314
2005-06	108	106	96	307
2006-07	108	56	130	294
2007-08	104	63	83	250
2008-09	84	83	86	253
2009-10	20	83	58	161
2010-11	40	121	62	223
2011-12	40	57	67	164
2012-13	20	128	59	207
2013-14	40	122	117	278

Tableau 7.6 – Synthèse des charges d'enseignement depuis 2004

J'ai également enseigné de manière ponctuelle en dehors de l'EMN. Le tableau 7.7 récapitule ces enseignements.

Cours	Années	Lieu (contact)	Niveau	Heures
Lagrangean Relaxation	2009	Univ. de los Andes (A. Medaglia)	M2, Ph.D	10h
Datamining	2007–2008	HEG Genève (E. Fragnière)	M2	12h
Programmation linéaire	2009–2011	HES-SO, Lausanne (S. Varone)	M2	18h
Planification	2004–2011	IMA Angers (E. Pinson)	M2	42h
Various topics	2008	Univ. of Sharjah (N. Brahimi)	M1/M2	12h
Recherche Opérationnelle	2010	Univ. de Bretagne Sud (M. Sevaux)	M1/M2	20h

Tableau 7.7 – Tableau Récapitulatif des cours donnés en dehors de l'EMN

7.7.2 Responsabilités pédagogiques

- J'ai exercé la responsabilité de l'option GOPL (Gestion des Opérations en Production et Logistique) à l'EMN de 2004-2009. L'option compte en moyenne 20 élèves par an. De 2004 à 2009, la responsabilité d'option s'étendait principalement sur la dernière année de programme, soit 450 heures à l'emploi du temps et un projet de fin d'étude de 6 mois. Mes principales tâches étaient d'assurer la promotion du programme et de recruter les élèves, de gérer l'emploi du temps et le recrutement d'intervenants extérieurs, d'assurer le suivi des élèves au cours de leur formation à l'école et pendant leur stage de fin d'études. Cette fonction a représenté de 80 à 100 UP selon les années.
- J'ai mené l'actualisation des programmes pédagogiques de l'option GOPL en 2006. Ce travail a été mis en oeuvre à partir de la rentrée de septembre 2006. Le programme ainsi défini a été suivi jusqu'en 2012.
- Responsabilité d'Unités de Valeur : le système d'Unités de valeur a été mis en place en 2006. Entre 2006 et 2009, au titre de responsable de l'option GOPL, j'étais responsable de 3 des 4 unités de valeur de dernière année. Mes responsabilités ont ensuite évolué, comme le montre le tableau 7.8.

7.7.3 Autres activités liées à l'enseignement

- Soutenances de stage opérateur : 27

Nom de l'UV	Années	Public	Heures	UP
Recherche Opérationnelle 2	2006–2009	M2	90	20
Management de la chaîne logistique	2006–2009	M2	90	20
Production et systèmes d'information	2006–2009	M2	90	20
Gestion des Opérations	2012–2014	M1	45	10
Gestion Industrielle	2010–2012	M1	90	20
Management des Opérations	2009–2012	M1	90	20
Operations Management	2012–2014	M1	65	10
Supply Chain Integrated Project	2013	M2	65	20

Tableau 7.8 – Responsabilités d'Unités de Valeurs depuis 2006

- Soutenances de missions courtes (études terrain, etc) : 3
- Soutenances de stages à l'étranger : environ 80
- Autres soutenances de projet : 8
- Jury de stage de diplôme HEG Genève : Cécile Zamorano (Givaudan) et Cyril Peyrot (Maison de Vessy), 2006.
- Session découverte Entreprises (3 jours de visite entreprises + 1/2 journée de débriefing) : 2011, 2012, 2013
- Salon Formathèque : 2 demi-journées
- Salon Europosgrads (Monterrey et Mexico, avril 2005)
- Voyage d'étude à l'Université de Sherbrooke (Canada) sur l'apprentissage par projets (1 semaine en octobre 2005)
- Jury pour les oraux du concours commun d'entrée aux Écoles de Mines 2011 et 2012 (5 jours)
- Intervention au Lycée Saint-Joseph (Bruz, 35) 4h en mai 2013
- Participation régulière au Comité pédagogique de l'EMN entre 2004 et 2009.

7.8 Autres activités

- Membre du Club Logistique des Pays de la Loire depuis 2013
- Membre du Groupe Projet Formation du pôle PASCA (Pôle Achats Supply Chain Atlantique)
- Membre du comité d'organisation du congrès 2008 de l'ASLOG (Association Française pour la Logistique) à Nantes.
- Mes activités d'enseignement (visite de stage, session découverte entreprise, visite organisées avec élèves, projets tutorés) et de valorisation m'ont amené à visiter près de 100 entreprises, majoritairement dans la région Pays de la Loire et dans le domaine de l'industrie ou des services.
- Par ailleurs, j'ai régulièrement participé à des salons professionnels dans le secteur de la logistique, du transport ou des systèmes d'information : Assises de la logistique, SITL, salons ERP, Progilog, Transports Public, Autonomic, salon de l'industrie (Nantes), congrès de l'ASLOG.

Bibliographie personnelle

A.1 Articles publiés dans des revues avec comité de lecture

1. A new consistent vehicle routing problem for the transportation of handicapped persons, *D. Feillet, Th. Garaix, F. Lehuédé, O. Péton, D. Quadri, Networks*, (2014) **63**, 3, 211–224,
2. A branch-and-cut-and-price approach for the pickup and delivery problem with shuttle routes, *R. Masson, S. Ropke, F. Lehuédé, O. Péton, European Journal of Operational Research* (2014) **236**,3, 849–862.
3. The dial-a-ride problem with transfers, *R. Masson, F. Lehuédé, O. Péton, Computers and Operations Research* (2014) **41**, 12–23
4. A multi-criteria large neighborhood search for the transportation of disabled people, *F. Lehuédé, R. Masson, S.N. Parragh, O. Péton, F. Tricoire, Journal of the Operational Research Society* (2014), **65**, 7, 983–1000
5. Efficient feasibility testing for request insertion in the pickup and delivery problem with transfers, *R. Masson, F. Lehuédé, O. Péton, Operations Research Letters* (2013) **41**, 211–215
6. An Adaptive Large Neighborhood Search for the Pickup and Delivery Problem with Transfers, *R. Masson, F. Lehuédé, O. Péton, Transportation Science* (2013) **47**, 3, 344–355
7. A DC programming heuristic applied to the logistics network design problem, *P.N. Thanh, N. Bostel, O. Péton, International Journal of Production Economics* (2012) **135**, 1, 94–105
8. A linear relaxation-based heuristic approach for logistics network design, *P.N. Thanh, O. Péton, N. Bostel, , Computers & Industrial Engineering* (2010) **59**, 4, 964–975
9. A dynamic model for facility location in the design of complex supply chains, *P.N. Thanh, N. Bostel, O. Péton, International Journal of Production Economics* (2008) **113**, 2, 678–693

A.2 Articles soumis dans des revues avec comité de lecture

1. Optimization of a city logistics transportation system with mixed passengers and goods, *R. Masson, A. Trentini, F. Lehuédé, N. Malhéné, O. Péton, H. Tlahig*, à paraître dans *EURO Journal on Transportation and Logistics*.

2. Sustainable supply chain network design : an optimization-oriented review, *M. Eskandarpour, P. Dejax, J. Miemczyk, O. Péton*, à paraître dans Omega.
3. Large neighborhood search heuristic for supply chain network design, *M. Eskandarpour, P. Dejax, O. Péton*, soumis à Computers and Operations Research (08/2014).
4. Optimization of customer orders routing in a collaborative distribution network, *F. Lehuédé, O. Péton, X. Tang* (10/2014).

A.3 Articles publiés dans des actes de congrès avec comité de lecture

1. A Large Neighborhood Search based heuristic for Supply Chain Network Design, *M. Eskandarpour, P. Dejax, O. Péton*, 5th International Conference on Information Systems, Logistics and Supply Chain (ILS 2014), Breda, Pays-Bas.
2. Simple Temporal Problems in Route Scheduling for the Dial-a-Ride Problem with Transfers, *R. Masson, F. Lehuédé, O. Péton*, 9th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR'12) (2012), Lecture Notes in Computer Science **7298**, 275–291 (2012) Nantes, France.
3. A shared passengers & goods city logistics system, *A. Trentini, R. Masson, F. Lehuédé, N. Malhéné, O. Péton, H. Tlahig*, 4th International Conference on Information Systems, Logistics and Supply Chain (ILS 2012), 10p (2012) Québec, Canada.
4. An Adaptive Large Neighborhood Search for the Pickup and Delivery Problem with Transfers, *R. Masson, F. Lehuédé, O. Péton*, ORP3, 10p (2011), Cadix, Espagne.
5. A tabu search algorithm for the dial-a-ride problem with transfers, *R. Masson, F. Lehuédé, O. Péton*, International Conference on Industrial Engineering and Systems Management (IESM 2011) (2011) Metz, France.
6. An adaptive large neighborhood search for the pickup and delivery problem with transfers, *R. Masson, F. Lehuédé, O. Péton*, Proceedings of the MIC 2011 Conference (2011) S3-09-1, Udine, Italie.
7. A linear relaxation-based heuristic for logistics network design, *P.N. Thanh, N. Bostel, O. Péton*, Conférence Internationale de Modélisation, Optimisation et Simulation des Systèmes (MOSIM) (2008) Paris, France.
8. Un système d'aide à la décision pour planifier les transports vers les établissements médico-sociaux, *F. Lehuédé, C. Pavageau, O. Péton*, 5ème conférence Handicap (2008) 168–173, Paris, France.
9. A dynamic model for the facility location in supply chain design, *P.N. Thanh, N. Bostel, O. Péton*, Proceedings of the EURO Winter Institute on Location and Logistics (2007) 402–418 Estoril, Portugal.

A.4 Congrès internationaux avec comité de sélection

1. The pickup and delivery problem with shuttle routes, *R. Masson, F. Lehuédé, O. Péton, S. Ropke*, Proceedings of TRISTAN VIII : Eight Triennial Symposium on Transportation Analysis, 4p, (2013) San Pedro de Atacama, Chili.
2. Optimization of a shared passengers and goods urban transportation network, *R. Masson, A. Trentini, F. Lehuédé, O. Péton, H. Tlahig, N. Malhéné*, Odysseus 2012, 4p, (2012) Mykonos, Grèce.

A.5 Autre conférences internationales

1. A mixed passengers-goods transportation network for territories with a low population density, *S. Gonguet, S. Charpentier, M. Jeannenot, F. Lehuédé, N. Bostel, O. Péton, B. Delahaie*, TRA 2014, Transport Research Arena, Paris, France.
2. A survey of optimization models and methods for sustainable supply chain network design, *M. Eskandarpour, P. Dejax, J. Miemczyk, O. Péton*, Isolde 2014, Naples/Capri, Italie.
3. Modeling of a supply chain network design problem in the construction sector ; *O. Péton, M.H. Hà, Th. Vincent*, Isolde 2014, Naples/Capri, Italie.
4. A three step decomposition approach for a transportation network design problem with non-linear costs, *F. Lehuédé, O. Péton*, International Network Optimization Conference (2013) Tenerife, Espagne.
5. Feasibility algorithms for two pickup and delivery problems with transfers, *F. Lehuédé, R. Masson, O. Péton*, 1st International workshop on synchronisation in transport, SynchroTrans 2013, Mainz, Allemagne.
6. Location of concentration warehouses in a pooled distribution network in the retail sector, *O. Péton, A. Brunel, F. Lehuédé*, EWGLA XX (2013), Ankara, Turquie.
7. An adaptive large neighborhood search for supply chain network design, *Eskandarpour, M., Dejax, P., Péton, O.*, EWGLA XX (2013), Ankara, Turquie.
8. An ALNS and route scheduling algorithms for the dial-a-ride problem with transfers, *F. Lehuédé, R. Masson, O. Péton*, VeRoLog 2012 (2012) Bologne, Italie.
9. A new time-consistent vehicle routing problem for the transportation of handicapped persons, *D. Feillet, Th. Garaix, F. Lehuédé, O. Péton, D. Quadri*, Network Optimization Workshop (NOW) (2010) Ajaccio, France.
10. The time-consistent vehicle routing problem, *D. Feillet, Th. Garaix, F. Lehuédé, O. Péton, D. Quadri*, Matheuristics (2010) Vienne, Autriche.
11. A tabu search based heuristic for the transportation of mentally disabled people, *F. Lehuédé, C. Pavageau, O. Péton*, ORAHS 2009 (2009) Leuven, Belgique.
12. Combined routing and staff scheduling model for home health care, *E.V. Gutiérrez, C.A. Amaya, Ch. Guéret, O. Péton, N. Velasco*, ORAHS 2009 (2009) Leuven, Belgique.
13. A Lagrangean relaxation based heuristic for logistics network design, *P.N. Thanh, N. Bostel, O. Péton*, EURO Working Group on Locational Analysis (2008) Elche, Espagne.
14. A linear relaxation heuristic for solving logistics network design and planning problems, *P.N. Thanh, N. Bostel, O. Péton*, ISSCM 08 : First International Symposium on Supply Chain Management (2008) Sharjah, Émirats Arabes Unis.
15. Designing a decision support system for the transportation of disabled persons, *O. Péton, P. Dejax, F. Lehuédé, M. Marhadour, C. Pavageau*, The 33rd international conference on Operational Research Applied to Health Services (ORAHS) (2007) Saint-Etienne, France.
16. An application of DC programming approach for a logistics network design problem, *P.N. Thanh, N. Bostel, O. Péton*, NCP 07 : The International Conference on NonConvex Programming (2007) Rouen, France.

A.6 Conférences nationales sans actes

1. Un état de l'art des modèles d'optimisation pour la conception des chaînes logistique durables, *M. Eskandarpour, P. Dejax, J. Miemczyk, O. Péton*, 5^{ème} journées de recherche SILOGIN, Novembre 2014, Nantes.

2. Prise en compte du développement durable dans les modèles et méthodes de conception des réseaux logistiques : un état de l'art, *O. Péton, M. Eskandarpour, P. Dejax, J. Miemczyk*, Journées de l'Optimisation, Mai 2014, Montréal, Canada.
3. Conception d'un réseau logistique mutualisé entre fournisseurs de la grande distribution, *J. Médina, F. Lehuédé, O. Péton*, ROADEF 2014, Bordeaux.
4. Mutualisation logistique : évaluation d'un réseau de distribution, *X. Tang, F. Lehuédé, O. Péton*, ROADEF 2014, Bordeaux.
5. Modélisation d'un problème de conception de chaîne logistique pour une entreprise de travaux publics, *O. Péton, M.H. Hà, Th. Vincent*, ROADEF 2014, Bordeaux.
6. A state of the art of supply chain design models and methods integrating the principles of sustainable development, *M. Eskandarpour, P. Dejax, O. Péton, N. Bostel*, ROADEF 2012, Angers.
7. Deux modèles de PLNE pour la planification de l'activité du personnel soignant pour l'hospitalisation à domicile, *A. Le Roux, Ch. Guéret, O. Péton*, ROADEF 2012, Angers.
8. Branch-and-cut-and-price pour un problème de collectes et livraisons avec transfert, *R. Masson, S. Ropke, F. Lehuédé, O. Péton*, ROADEF 2012, Angers.
9. Optimisation d'un système de transport urbain mixte passagers/marchandises, *R. Masson, A. Trentini, F. Lehuédé, N. Malhéné, O. Péton, H. Tlahig*, ROADEF 2012, Angers
10. Une recherche à voisinage large pour un problème multicritère de transport à la demande, *F. Lehuédé, R. Masson, S. Parragh, O. Péton, F. Tricoire*, ROADEF 2012, Angers.
11. Etude d'un problème de transport à la demande avec transfert, *R. Masson, F. Lehuédé, O. Péton*, ROADEF 2010, Toulouse
12. Une heuristique pour le problème de tournées de véhicules régulières, *D. Feillet, Th. Garaix, F. Lehuédé, O. Péton, D. Quadri*, ROADEF 2010, Toulouse.
13. Modèle et algorithme pour la reconfiguration des réseaux logistiques complexes, *N. Bostel, P.N. Thanh, O. Péton*, Journées de l'optimisation, mai 2009, Montréal, Canada.
14. Comparaison de trois heuristiques pour la planification stratégique de réseaux logistiques, *P.N. Thanh, N. Bostel, O. Péton*, ROADEF 2009, Nancy.
15. Conception de Tournées de Véhicules Régulières, *D. Quadri, Th. Garaix, D. Feillet, F. Lehuédé, O. Péton*, ROADEF 2009, Nancy.
16. Optimisation de la planification de bourses d'échanges de technologies, *Ch. Guéret, O. Morineau, C. Pavageau, O. Péton, D. Poncelet*, ROADEF 2009, Nancy
17. Un système d'aide à la décision pour le transport collectif de personnes handicapées, *O. Péton, P. Dejax, F. Lehuédé, C. Pavageau, M. Marhadour*, ROADEF 2008, Clermont-Ferrand
18. Une heuristique basée sur la relaxation linéaire pour la planification de réseaux logistiques complexes, *P.N. Thanh, N. Bostel, O. Péton*, ROADEF 2008, Clermont-Ferrand
19. Un modèle dynamique pour la conception stratégique d'un réseau logistique, *P.N. Thanh, N. Bostel, O. Péton*, ROADEF 2007, Grenoble

A.7 Rapports de recherche

1. Planification de l'activité du personnel soignant pour l'hospitalisation à domicile, *A. Le Roux, Ch. Guéret, O. Péton*, Rapport de Recherche 12/2/AUTO (2012), 24 pages.

2. A Lagrangean heuristic for a logistics network design problem, *P.N. Thanh, N. Bostel, O. Péton*, Rapport de Recherche EMN (2011), 24 pages.
3. A route feasibility algorithm for the Dial A Ride Problem with Transfers, *R; Masson, F. Lehuédé, O. Péton*, Rapport de Recherche EMN 10/2/AUTO (2010), 19 pages.

A.8 Autres publications nationales

1. Outil d'optimisation pour le transport des personnes handicapées mentales, *F. Lehuédé, C. Pavageau, O. Péton*, Techniques de l'ingénieur, RE128 (2009)

A.9 Conférences invitées

1. *Optimization of intermodal transport systems : some lessons learned*, Urban Operations Research workshop, Nanzan University, Nagoya, Japon (13-14 décembre 2014).
2. *Design of a collaborative distribution network*, Séminaire, Nanzan University, Nagoya, Japon (11 décembre 2014).
3. Demand responsive transport problems with passengers transfers : models and methods, Seminarios CIO, Université de Lisbonne (mars 2012).
4. A decision support system for the transportation of disabled persons, Seminario Departamento de Ingenieria Industrial, Université des Andes, Bogota, Colombie (avril 2009).
5. Conception et planification stratégiques des chaînes logistiques, Lunch Recherche, HEG Genève (2006).

Bibliographie

- AKÇALI, E., ÇETINKAYA, S. et ÜSTER, H. (2009). Network design for reverse and closed-loop supply chains : an annotated bibliography of models and solution approaches. *Networks*, 53(3): 231–248.
- AKSEN, D., OZYURT, Z. et ARAS, N. (2007). Open vehicle routing problem with driver nodes and time deadlines. *Journal of the Operational Research Society*, 58(9):1223–1234.
- APPLEGATE, D. et COOK, W. (1991). A computational study of the job-shop scheduling problem. *ORSA Journal on Computing*, 3(2):149–156.
- ARAS, N., BOYACI, T. et VERTER, V. (2010). Designing the reverse logistics network. In FERGUSON, M. et SOUZA, G., éditeurs : *Closed Loop Supply Chains : New Developments to Improve the Sustainability of Business Practices*, pages 67–98. Taylor and Francis, CRC Press.
- ATAHRAN, A., LENTÉ, C. et T’KINDT, V. (2014). A multicriteria dial-a-ride problem with an ecological measure and heterogeneous vehicles. *Journal of Multi-Criteria Decision Analysis*, page doi : 10.1002/mcda.1518.
- AVELLA, P., BOCCIA, M., SFORZA, A. et VASIL’EV, I. (2009). An effective heuristic for large-scale capacitated facility location problems. *Journal of Heuristics*, 15(6):597–615.
- BEAMON, B. M. (1998). Supply chain design and analysis : models and methods. *International Journal of Production Economics*, 55(3):281–294.
- BERTACCO, L., FISCHETTI, M. et LODI, A. (2007). A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization*, 4:63–76.
- BRANDÃO, J. (2004). A tabu search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 157:552–564.
- CAISSE NATIONALE DE SOLIDARITÉ POUR L’AUTONOMIE (2009). Rapport du groupe de travail frais de transport des personnes en situation de handicap. <http://www.asea43.org/pdf/juridique1/handicape/rapports/rapportfraisdetransports.pdf>. (accédé le 20 avril 2014).
- COPADO-MÉNDEZ, P. J., BLUM, C., GUILLÉN-GOSÁLBEZ, G. et JIMÉNEZ, L. (2013). Large neighbourhood search applied to the efficient solution of spatially explicit strategic supply chain management problems. *Computers & Chemical Engineering*, 49:114–126.
- CORDEAU, J.-F. et LAPORTE, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B : Methodological*, 37(6):579 – 594.

- CORDEAU, J.-F. et LAPORTE, G. (2005). Tabu search heuristics for the vehicle routing problem. In SHARDA, R., VOSS, S., REGO, C. et ALIDAEE, B., éditeurs : *Metaheuristic Optimization via Memory and Evolution*, volume 30 de *Operations Research/Computer Science Interfaces Series*, pages 145–163. Springer US.
- CORTÉS, C. E., MATAMALA, M. et CONTARDO, C. (2010). The pickup and delivery problem with transfers : Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3):711–724.
- CRAINIC, T. G., ERRICO, F., REI, W. et RICCIARDI, N. (2012). Integrating C2E and C2C traffic into city logistics planning. *Procedia - Social and Behavioral Sciences*, 39(0):47–60. Seventh International Conference on City Logistics, June 7–9,2011, Mallorca, Spain.
- CRAINIC, T. G., PERBOLI, G., MANCINI, S. et TADEI, R. (2010). Two-echelon vehicle routing problem : a satellite location analysis. *Procedia - Social and Behavioral Sciences*, 2(3):5944–5955.
- DASKIN, M. S., SNYDER, L. V. et BERGER, R. (2005). Facility location in supply chain design. In LANGEVIN, A. et RIOPEL, D., éditeurs : *Logistics systems : design and optimization*, pages 39–65. Springer US.
- DECHTER, R., MEIRI, I. et PEARL, J. (1991). Temporal constraint networks. *Artificial Intelligence*, 49(1-3):61–95.
- DESAULNIERS, G., DESROSIERS, J. et SOLOMON, M., éditeurs (2005). *Column generation*. GERAD 25th Anniversary Series. Springer.
- ESKANDARPOUR, M., DEJAX, P., MIEMCZYK, J. et PÉTON, O. (2015). Sustainable supply chain network design : an optimization-oriented review. *Omega*, 54:11–32.
- ESKANDARPOUR, M., DEJAX, P. et PÉTON, O. (2014a). A large neighborhood search based heuristic for supply chain network design. In *Proceedings of the ILS 2014 conference, Breda, The Netherlands*.
- ESKANDARPOUR, M., DEJAX, P. et PÉTON, O. (2014b). A large neighborhood search heuristic for supply chain network design. *Computers & Operations Research (soumis)*.
- FEILLET, D. (2010). A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR, A Quarterly Journal of Operations Research*, 8(4):407–424.
- FEILLET, D., DEJAX, P., GENDREAU, M. et GUÉGUEN, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints : Application to some vehicle routing problems. *Networks*, 44:216–229.
- FEILLET, D., GARAIX, T., LEHUÉDÉ, F., PÉTON, O. et QUADRI, D. (2014). A new consistent vehicle routing problem for the transportation of people with disabilities. *Networks*, 63(3):211–224.
- FISCHETTI, M., GLOVER, F. et LODI, A. (2005). The feasibility pump. *Mathematical Programming*, 104(1):91–104.
- FLESZAR, K., OSMAN, I. H. et HINDI, K. S. (2009). A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 195(3):803–809.
- FU, Z., EGGLESE, R. et LI, L. Y. O. (2005). A new tabu search heuristic for the open vehicle routing problem. *Journal of the Operational Research Society*, 56(3):267–274.
- GOLDEN, B., RAGHAVAN, S. et WASIL, E. (2008). *The Vehicle Routing Problem : Latest Advances and New Challenges : latest advances and new challenges*. Operations Research/Computer Science Interfaces Series. Springer.

- GRABISCH, M. (1997). Alternative representations of discrete fuzzy measures for decision making. *International Journal of Uncertainty, Fuzziness, and Knowledge Based Systems*, 5:587–607.
- GRABISCH, M. et LABREUCHE, C. (2010). A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research*, 175:247–286.
- GROËR, C., GOLDEN, B. et WASIL, E. (2009). The consistent vehicle routing problem. *Manufacturing & Service Operations Management*, 11(4):630–643.
- GUASTAROBA, G. et SPERANZA, M. (2012). Kernel search for the capacitated facility location problem. *Journal of Heuristics*, 18(6):877–917.
- GUIGNARD, M. (2003). Lagrangean relaxation. *TOP : An Official Journal of the Spanish Society of Statistics and Operations Research*, 11(2):151–200.
- HAKIMI, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459.
- HEMMELMAYR, V. C., CORDEAU, J.-F. et CRAINIC, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12):3215–3228.
- HINOJOSA, Y., KALCSICS, J., NICKEL, S., PUERTO, J. et VELTEN, S. (2008). Dynamic supply chain design with inventory. *Computers & Operations Research*, 35(2):373–391.
- JEPSEN, M., PETERSEN, B., SPOORENDONK, S. et PISINGER, D. (2008). Subset-row inequalities applied to the vehicle routing problem with time windows. *Operations Research*, 56(2):497–511.
- KEENEY, R. et RAIFFA, H. (1976). *Decision with Multiple Objectives*. Wiley, New York.
- KERIVIN, H. L. M., LACROIX, M., MAHJOUR, A. R. et QUILLIOT, A. (2008). The splittable pickup and delivery problem with reloads. *European Journal of Industrial Engineering*, 2(2):112–133.
- KOHL, N., DESROSIERS, J., MADSEN, O. B. G., SOLOMON, M. M. et SOUMIS, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33:101–116.
- LAPIERRE, S. D., RUIZ, A. B. et SORIANO, P. (2004). Designing distribution networks : formulations and solution heuristic. *Transportation Science*, 38:174–187.
- LE THI, H. A. (1997). Contribution à l’optimisation non convexe et l’optimisation globale : théorie, algorithmes et applications. *Habilitation à Diriger des Recherches, Université de Rouen, France*.
- LE THI, H. A. et PHAM, D. T. (2005). The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, 133(1-4):23–46.
- LE THI, H. A., PHAM, D. T. et LE, D. M. (1999). Exact penalty in DC programming. *Vietnam Journal of Mathematics*, 27:169–178.
- LEHUÉDÉ, F., MASSON, R., PARRAGH, S. N., PÉTON, O. et TRICOIRE, F. (2014). Multi-criteria lns for transportation of disabled people. *Journal of the Operational Research Society*, 65(7):983–1000.
- LEHUÉDÉ, F., PAVAGEAU, C. et PÉTON, O. (2008). Un système d’aide à la décision pour planifier les transports vers les établissements médico-sociaux. *In Actes de la 5ème conférence Handicap, Paris*.
- LEHUÉDÉ, F., PAVAGEAU, C. et PÉTON, O. (2009a). Outil d’optimisation pour le transport des personnes handicapées mentales. *Techniques de l’ingénieur*, RE128.

- LEHUÉDÉ, F., PAVAGEAU, C. et PÉTON, O. (2009b). A tabu search based heuristic for the transportation of mentally disabled people. *In Proceedings of the ORAHS 09 conference, Leuven, Belgium.*
- LETCHFORD, A. N., LYSGAARD, J. et EGGLESE, R. W. (2007). A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society*, 58(12):1642–1651.
- LI, F., GOLDEN, B. et WASIL, E. (2007). The open vehicle routing problem: algorithms, large-scale test problems, and computational results. *Computers & Operations Research*, 34:2918–2930.
- LI, X. et TIAN, P. (2006). An ant colony system for the open vehicle routing problem. *Lecture Notes in Computer Science*, 4150:356–363.
- LIN, C. K. Y. (2008). A cooperative strategy for a vehicle routing problem with pickup and delivery time windows. *Computers & Industrial Engineering*, 55(4):766–782.
- LIU, R. et JIANG, Z. (2012). The close-open mixed vehicle routing problem. *European Journal of Operational Research*, 220(2):349–360.
- LYSGAARD, J. (2003). CVRPSEP : a package of separation routines for the capacitated vehicle routing problem. Working paper 03-04, Department of Management Science and Logistics, Faculty of Business Administration, the Aarhus School of Business.
- MASSON, R. (2012). *Problèmes de collectes et livraisons avec transferts*. Thèse de doctorat, École Centrale de Nantes.
- MASSON, R., LEHUÉDÉ, F. et PÉTON, O. (2013a). An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355.
- MASSON, R., LEHUÉDÉ, F. et PÉTON, O. (2013b). Efficient feasibility testing for request insertion in the pickup and delivery problem with transfers. *Operations Research Letters*, 41(3):211–215.
- MASSON, R., LEHUÉDÉ, F. et PÉTON, O. (2014a). The dial-a-ride problem with transfers. *Computers & Operations Research*, 41(0):12–23.
- MASSON, R., LEHUÉDÉ, F. et PÉTON, O. (2012). Simple temporal problems in route scheduling for the dial-a-ride problem with transfers. *In* BELDICEANU, N., JUSSIEN, N. et PINSON, E., éditeurs : *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 7298 de *Lecture Notes in Computer Science*, pages 275–291.
- MASSON, R., ROPKE, S., LEHUÉDÉ, F. et PÉTON, O. (2014b). A branch-and-cut-and-price approach for the pickup and delivery problem with shuttle routes. *European Journal of Operational Research*, 236(3):849–862.
- MASSON, R., TRENTINA, A., LEHUÉDÉ, F., MALHÉNÉ, N., PÉTON, O. et TLAHIG, H. (2014c). Optimization of a city logistics transportation system with mixed passengers and goods. *EURO Journal on Transportation and Logistics*, submitted.
- MEIXELL, M. J. et GARGEYA, V. B. (2005). Global supply chain design : a literature review and critique. *Transportation Research Part E : Logistics and Transportation Review*, 41(6):531–550.
- MELO, M. T., NICKEL, S. et Saldanha-da GAMA, F. (2009). Facility location and supply chain management - a review. *European Journal of Operational Research*, 196(2):401–412.
- MITROVIĆ-MINIĆ, S. et LAPORTE, G. (2006). The pickup and delivery problem with time windows and transshipment. *INFOR*, 44(3):217–228.
- OWEN, S. H. et DASKIN, M. S. (1998). Strategic facility location : a review. *European Journal of Operational Research*, 111(3):423–447.

- PAQUETTE, J., CORDEAU, J.-F. et LAPORTE, G. (2009). Quality of service in dial-a-ride operations. *Computers & Industrial Engineering*, 56(4):1721–1734.
- PAQUETTE, J., CORDEAU, J.-F., LAPORTE, G. et PASCOAL, M. M. (2013). Combining multi-criteria analysis and tabu search for dial-a-ride problems. *Transportation Research Part B : Methodological*, 52(0):1–16.
- PARRAGH, S. (2009). *Ambulance routing problems with rich constraints and multiple objectives*. Thèse de doctorat, University of Vienna, Austria.
- PATI, R., JANS, R. et TYAGI, R. (2013). Green logistics network design : A critical review. In *24th Annual POMS conference*, Denver, USA.
- PERBOLI, G., TADEI, R. et VIGO, D. (2011). The two-echelon capacitated vehicle routing problem : models and math-based heuristics. *Transportation Science*, 45(3):364–380.
- PHAM, D. T. et EL BERNOUSSI, S. (1988). Duality in D.C. (difference of convex functions) optimization. subgradient methods. In HOFFMANN, K.-H., ZOWE, J., HIRIART-URRUTY, J.-B. et LEMARÉCHAL, C., éditeurs : *Trends in Mathematical Optimization*, volume 84 de *International Series of Numerical Mathematics*, pages 277–293. Birkhauser Basel.
- PIPAME (2011). Pratiques de logistique collaborative : quelles opportunités pour les PME/ETI ? <http://www.dgcis.gouv.fr/etudes-et-statistiques/catalogue-prospective?page=1>. (accédé le 18 avril 2014).
- PISINGER, D. et ROPKE, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435.
- PRESCOTT-GAGNON, E., DESAULNIERS, G. et ROUSSEAU, L.-M. (2009). A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks*, 54(4):190–204.
- PRODHON, C. et PRINS, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17.
- PÉTON, O., DEJAX, P., LEHUÉDÉ, F., PAVAGEAU, C. et MARHADOUR, M. (2008). Un système d’aide à la décision pour le transport collectif de personnes handicapées. In *Actes de la conférence ROADEF 2008, Clermont-Ferrand*.
- PÉTON, O., HÀ, M. H. et VINCENT, T. (2014a). Modeling of a supply chain network design problem in the construction sector. In *Proceedings of the ISOLDE 2014 conference, Naples/Capri, Italy*.
- PÉTON, O., HÀ, M. H. et VINCENT, T. (2014b). Modélisation d’un problème de conception de chaîne logistique pour une entreprise de travaux publics. In *Conférence ROADEF 2014, Bordeaux, France*.
- QU, Y. et BARD, J. F. (2012). A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research*, 39(10):2439–2456.
- RAIS, A., ALVELOS, F. et CARVALHO, M. (2014). New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235(3):530–539.
- REKIEK, B., DELCHAMBRE, A. et SALEH, H. A. (2006). Handicapped person transportation : an application of the grouping genetic algorithm. *Engineering Applications of Artificial Intelligence*, 19(5):511–520.
- REPOUSSIS, P., TARANTILIS, C. D. et IOANNOU, G. (2007). The open vehicle routing problem with time windows. *Journal of the Operational Research Society*, 58(3):355–367.

- ROPKE, S. et PISINGER, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40:455–472.
- SAHIN, G. et SÜRAL, H. (2007). A review of hierarchical facility location models. *Computers & Operations Research*, 34(8):2310–2331.
- SAVELSBERGH, M. W. P. (1992). The vehicle routing problem with time windows : minimizing route duration. *ORSA Journal on Computing*, 4(2):146–154.
- SCHRIMPF, G., SCHNEIDER, J., STAMM-WILBRANDT, H. et DUECK, G. (2000). Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159:139–171.
- SHANG, J. S. et CUFF, C. K. (1996). Multicriteria pickup and delivery problem with transfer opportunity. *Computers & Industrial Engineering*, 30(4):631–645.
- SHAPLEY, L. S. (1953). A value for n -person games. Contribution to the Theory of Games. *Annals of Mathematics Studies*, 2.
- SHAW, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming*, pages 417–431.
- SHOSTAK, R. (1981). Deciding linear inequalities by computing loop residues. *Journal of the Association for Computing Machinery*, 28:769–779.
- SPADA, M., BIERLAIRE, M. et LIEBLING, T. M. (2005). Decision-aiding methodology for the school bus routing and scheduling problem. *Transportation Science*, 39:477–490.
- TANG, X., LEHUÉDÉ, F. et PÉTON, O. (2014). Mutualisation logistique : évaluation d'un réseau de distribution. In *Conférence ROADEF 2014, Bordeaux, France*.
- TARJAN, R. E. (1981). Shortest paths. Rapport technique, AT&T Bell Laboratories, Murray Hill, NJ.
- THANH, P., BOSTEL, N. et PÉTON, O. (2011). A Lagrangean heuristic for a logistics network design problem. Rapport de recherche, Ecole des Mines de Nantes.
- THANH, P. N. (2008). *Conception et planification stratégique des réseaux logistiques*. Thèse de doctorat, Université de Nantes.
- THANH, P. N., BOSTEL, N. et PÉTON, O. (2008). A dynamic model for facility location in the design of complex supply chains. *International Journal of Production Economics*, 113(2):678–693.
- THANH, P. N., BOSTEL, N. et PÉTON, O. (2012). A DC programming heuristics applied to logistics network design problem. *International Journal of Production Economics*, 135(1):94–105.
- THANH, P. N., PÉTON, O. et BOSTEL, N. (2010). A linear relaxation-based heuristic for logistics network design. *Computers & Industrial Engineering*, 59:964–975.
- TOLAND, J. (1978). Duality in non convex optimization. *Journal of Mathematical Analysis and Applications*, 66:399–415.
- TOTH, P. et VIGO, D. (1997). Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, 31(1):60–71.
- TOTH, P. et VIGO, D. (2002). *The vehicle routing problem*. SIAM monographs on discrete mathematics and applications. Society for Industrial and Applied Mathematics.

- TRENTINI, A. (2012). *Proposition d'un système de transport urbain mixte : application dans le cadre de la ville moyenne de La Rochelle*. Thèse de doctorat, Paris ENMP.
- TRENTINI, A. et MALHÉNE, N. (2010). Toward a shared urban transport system ensuring passengers and goods cohabitation. *TeMa - Trimestrale del Laboratorio Territorio Mobilità Ambiente*.
- TRICOIRE, F. (2012). Multi-directional local search. *Computers & Operations Research*, 39(12): 3089–3101.
- VINCENT, T., SEIPP, F., RUZIKA, S., PRZYBYLSKI, A. et GANDIBLEUX, X. (2013). Multiple objective branch and bound for mixed 0-1 linear programming : corrections and improvements for the biobjective case. *Computers and Operations Research*, 40(1):498–509.
- WEBER, A. et PEIK, G. (1909). *Über den Standort der Industrie*. J.C.B. Mohr (Paul Siebeck).

Sélection d'articles

- A dynamic model for facility location in the design of complex supply chains, *P.N. Thanh, N. Bostel, O. Péton, International Journal of Production Economics* (2008) **113**, **2**, 678–693.
- A DC programming heuristic applied to the logistics network design problem, *P.N. Thanh, N. Bostel, O. Péton, International Journal of Production Economics* (2012) **135**, **1**, 94–105.
- A new consistent vehicle routing problem for the transportation of handicapped persons, *D. Feillet, Th. Garaix, F. Lehuédé, O. Péton, D. Quadri, Networks*, (2014) **63**, **3**, 211–224.
- A multi-criteria large neighborhood search for the transportation of disabled people, *F. Lehuédé, R. Masson, S.N. Parragh, O. Péton, F. Tricoire, Journal of the Operational Research Society* (2014), **65**, 983–1000.
- A branch-and-cut-and-price approach for the pickup and delivery problem with shuttle routes, *R. Masson, S Ropke, Lehuédé, O. Péton, European Journal of Operational Research* (2014) **236**,**3**, 849–862.

A dynamic model for facility location in the design of complex supply chains

Phuong Nga Thanh^a, Nathalie Bostel^b, Olivier Péton^{a,*}

^a*École des Mines de Nantes, IRCCyN 4 rue Alfred Kastler, 44307 Nantes Cedex 3, France*

^b*IUT de Saint-Nazaire, IRCCyN, 58 rue Michel Ange, B.P. 420, 44606 Saint-Nazaire Cedex, France*

Received 23 December 2006; accepted 19 October 2007

Available online 15 February 2008

Abstract

This paper proposes a mixed integer linear program (MILP) for the design and planning of a production–distribution system. This study aims to help strategic and tactical decisions: opening, closing or enlargement of facilities, supplier selection, flows along the supply chain. These decisions are dynamic, i.e. the value of the decision variables may change within the planning horizon. The model considers a multi-echelon, multi-commodity production–distribution network with deterministic demands. We present one application: how to plan the expansion of a company that has to face increasing demands. We report numerical experiments with an MILP solver.

© 2008 Elsevier B.V. All rights reserved.

Keywords: Supply chain design; Facility location; Strategic planning; Dynamic model; Mixed integer linear programming

0. Introduction

Locating facilities, designing the supply chain, selecting suppliers, defining the physical and information flows along a supply chain are critical parts of corporate strategic planning. These long-term decisions influence the activities of the company, and require a deep knowledge of the company and its environment.

Companies can be faced with facility location and supply chain design problems in different situations.

Let us mention a few examples:

- A company that extends its activity to new geographical areas has to locate facilities and design a new part of its supply chain.
- When the activity approaches the limits of the production capacity, the choice between opening new facilities or enlarging existing ones is a real strategic issue.
- In the case of an obsolete facility, is it better to renovate it or close it and replace it by a new one?
- When demand is declining, restructuring the supply chain involves choosing where the activity should be reduced.
- The merger of two companies involves merging the supply chains too.

All of these examples underline the importance of long-term planning in supply chain design.

*Corresponding author. Tel.: +33 251 858 313;

fax: +33 251 858 349.

E-mail addresses: phuong-nga.thanh@emn.fr (P.N. Thanh),
Nathalie.Bostel@univ-nantes.fr (N. Bostel),
olivier.peton@emn.fr (O. Péton).

The objective of this paper is to propose a mathematical model for designing and optimising a supply chain over a multi-period horizon. The model includes important features such as the location of several types of facilities, supplier selection, multiple finished goods with their respective bills of materials, fixed and variable costs. Human and organisational factors are beyond the scope of the present paper which only models the quantitative aspects of the problems. Our purpose is to provide managers with quantitative tools for decision-making, but any real-life decision process should consider both qualitative and quantitative information.

The paper is organised as follows. A literature review is presented in Section 1 while the mathematical model is defined in Section 2. Computational results are reported in Section 3. Some discussion and further research are presented in the conclusion.

1. Problem context and related studies

1.1. Facility location and supply chain design

Facility location and strategic planning have been the subject of a considerable number of publications in the fields of management science and operational research. The first researches focused on single facility and then on multiple facility location. New constraints were appended to the existing models, in order to yield more realistic ones.

During the previous decade, many works have addressed the dynamic location problem. In this class of problems, the decisions are spread out over a long-term planning horizon, and the decision variables are time-dependent. The purpose is to minimise or maximise an objective function over this horizon. For example, how should a growing company develop its logistic system in the five forthcoming years so that the demand and the logistic constraints remain satisfied at a minimum cost?

Dynamic location problems were first described in a single facility location context. In the first paper in this field (Ballou, 1968), the objective was to locate one warehouse in order to maximise profits over a finite planning horizon. The author solves a series of static location problems (one for each time period) and then resorts to dynamic programming. The problem was extended to multiple facilities by Scott (1971).

There is such a large variety of company logistic networks that it seems unrealistic to build a completely generic model that would adapt to any organisation. The most recent models include many additional features, with the idea of reflecting some real cases or of focusing on some precise aspects of location problems. However, no published paper integrates all the possible features simultaneously. Among the main characteristics in the recent models, we can list the following:

- Supply chain with multiple echelons: suppliers, plants, warehouses, customers, etc.
- Multiple products or families of products.
- Complex product flows: several layers of plants and warehouses. exchange of products between plants or warehouses, direct deliveries to some customers, reverse logistics, re-manufacturing, etc.
- Choice between several transportation modes, several production technologies.
- Variety of constraints: capacity constraints, budget constraints, etc.
- Complex structure of the costs: fixed and variable costs, linear or nonlinear costs.
- Hybrid strategic/tactical models with inventories: average inventory, safety stock, cyclic inventories.

Our assumptions and selected features are detailed in Section 2.

1.2. Literature review

Instead of reviewing all the related works, we present five papers that helped us to build our own model, and that cover the main characteristics of location problems. For a detailed review of the field, interested readers should refer to the following: Daskin et al. (2005) propose an extensive review of location problems, Beamon (1998) distinguishes models with deterministic data from those with stochastic data, Owen and Daskin (1998) clearly separate the static and dynamic models.

Cordeau et al. (2006) propose a static model considering a multi-commodity, multi-facility and single-country network. The decision variables concern the number of locations, the capacity and technology of manufacturing in plants and warehouses, selection of suppliers, selection of distribution channels, transportation modes and material flows. Bills of materials are also taken into

consideration. To solve this problem, the authors present two methods: a simplex-based branch-and-bound approach and a Benders decomposition approach. They use some valid inequalities to strengthen the LP relaxation. In our formulation, we take a similar structure network but put the model in a multi-period horizon.

Hamer-Lavoie and Cordeau (2006) simplify this model by removing the suppliers and the bills of materials. They suppose that the location has already been chosen for plants and the model focuses on warehouse location. The model is dynamic with stochastic demands, and takes inventories into account, including the safety stock. The authors propose a linear approximation for the last constraints concerning safety stock and a branch-and-bound method strengthened by valid inequalities. The main contributions of this paper are the introduction of stochastic demands and safety stock management.

Dias et al. (2006) work on the re-engineering of a two-echelon network (facilities and customers). The authors suppose that facilities can be opened, closed and reopened more than once during the planning horizon. They study these conditions within three scenarios: with maximum capacity restrictions; with both maximum and minimum capacity restrictions; and with a maximum capacity that decreases. All of these problems are solved by primal-dual heuristics. In this paper, three linear formulations correspond to the three previous scenarios and their linear dual formulations are presented. The main characteristic of this model is its great flexibility. The idea that we have taken from this paper is the possibility of opening and closing a facility during the planning horizon. Nevertheless, we forbid a closed facility to be reopened.

With the same flexibility idea, Melo et al. (2006) aim at relocating the network with expansion/reduction capacity scenarios. Capacity can be exchanged between an existing facility and a new one, or between two existing facilities under some conditions. Each change of capacity is penalised by a cost. In this model, closed facilities cannot be reopened, and new facilities will remain in activity until the end of the planning horizon. Some budget constraints are also added, which is rare in the literature. This proposition suggested to us the introduction of the modular capacity for each facility.

Vila et al. (2006) propose a dynamic model in a much more specialised context. They consider an

application in the lumber industry, but their model can be applied to other sectors. The authors consider an international network, with deterministic demands. They consider external suppliers, capacitated plants and warehouses, the choice between a set of available technologies, the possibility of adding capacity options to the facilities, and a list of substitute products to replace standard ones. They define three levels of storage. Their objective function is the after-tax profit, which has to be maximised.

1.3. Summary

To complement the presentation of the most relevant papers in this field, Table 1 sums up the characteristics of 15 models from the literature of the last decade. The first four references concern static models while the last eleven are dynamic. Refs. [1–9] deal with only the strategic level and do not handle inventories, whereas the last six cover both strategic and tactical decision levels. Modular capacities mean that the capacities may vary, due for example to supplementary options of capacity that may be added to the existing facilities. In the column Closing/reopening, we do not distinguish models that authorise or forbid the reopening of a facility after closure. Resolution methods for each model are reported in Table 2. We classify these methods in four categories: exact methods, commercial solvers, heuristic methods and meta-heuristics.

2. Problem formulation

2.1. Problem settings and description

We consider the multi-period, multi-commodity multi-facility location problem with deterministic customer demands as the core component of the more general supply chain design and planning problem. We consider a network like the one depicted in Fig. 1, with four main layers composed of suppliers, plants, warehouses and customers. The bill of materials is also taken into consideration.

In the next sections, we use the term “facility” for plants and warehouses when we describe a common characteristic of both. For other cases, we distinguish plants (that are dedicated to production activity) and warehouses (that are dedicated to storage activity). In the same way, we distinguish “treatment” and “storage” activity in a warehouse

Table 1
Review of existing models

	Static	Dynamic	Inventories	Deter- ministic	Stochastic	Modular capacities	Mono- echelon	Multi- echelon	Closing/ reopening	External suppliers	Bill of materials	Transpor- tation	International network	Relocation	Budget constraint
Cordeau et al. (2006)	x			x				x		x	x	x			
Filho and Galvão (1998)	x			x			x								
Paquet et al. (2004)	x			x	x			x		x	x				
Pirkul and Jayaraman (1998)	x			x				x							
Canel et al. (2001)		x		x				x	x			x			
Dias et al. (2006)		x		x	x		x		x						
Hinojosa et al. (2000)		x		x				x	x						
Melachrinoudis and Min (2000)		x		x	x			x	x					x	x
Saldanha da Gama and Captivo (1998)		x		x			x		x						x
Ambrosino and Scutellà (2005)		x	x	x				x				x			
Hamer-Lavoie and Cordeau (2006)		x	x		x			x				x			
Martel (2005)		x	x	x		x		x	x	x	x		x		
Melo et al. (2006)		x	x	x		x		x	x	x				x	x
Syam (2002)		x	x	x				x				x			
Vila et al. (2006)		x	x	x		x		x		x			x		

Table 2
Review of existing methods

	Exact methods	Commercial solvers	Heuristic methods	Metaheuristics
Cordeau et al. (2006)	x	x		
Filho and Galvão (1998)	x			x
Paquet et al. (2004)	x	x		
Pirkul and Jayaraman (1998)	x		x	
Canel et al. (2001)	x			
Dias et al. (2006)			x	
Hinojosa et al. (2000)	x		x	
Melachrinoudis and Min (2000)		x		
Saldanha da Gama and Captivo (1998)	x		x	
Ambrosino and Scutellà (2005)		x		
Hamer-Lavoie and Cordeau (2006)	x	x		
Martel (2005)	x	x		
Melo et al. (2006)		x		
Syam (2002)				x
Vila et al. (2006)		x		

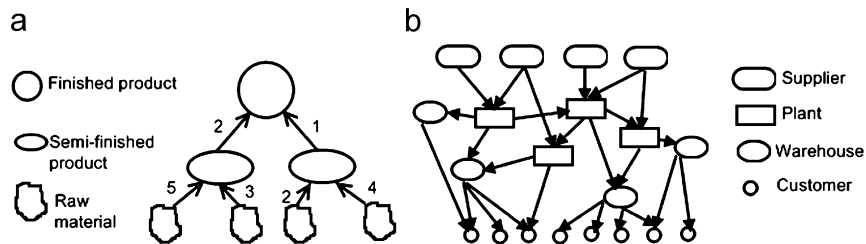


Fig. 1. Example of the bill of materials (a) and the general network (b).

in the sense that “treatment” consists of all actions to place and remove a product while “storage” means that the warehouse has to hold a product. As regards plants, the term “treatment” can be understood as a production process.

Our purpose is to make strategic decisions concerning these actors for each period of the planning horizon: selection of suppliers; opening/closing of facilities; planning capacity for existing facilities; production management and distribution management. Inventory management is included in the model even though it is a tactical decision, as it influences strategic decisions such as capacity planning.

We consider a network with some existing plants and warehouses that may all be closed. In addition, there is a set of potential locations, where a new plant or a new warehouse can be opened. We assume that the status of a facility changes no more than once during the planning period. For the initial

period, we suppose that all locations and suppliers have already been chosen.

We adopt a flexible structure for the network: products can be transferred between plants or delivered directly from plants to important customers. We do not consider transfers between warehouses, but such a modification requires only minor changes to the model.

Each supplier has a restriction on the available raw materials and a limited capacity. The supplier will offer a discount on the fixed cost if it is selected to deliver at least two raw materials. The model has to select suppliers in general and then decide which supplier will deliver which raw material in detail.

Considering the planning capacity, each site has an initial capacity as well as a limited maximal installable capacity. For example, a warehouse is built on a plot of 26 000 m². At the beginning, 20 000 m² are used for the building and 6000 m² remain available for possible extension. In this

example, the initial capacity is 20 000 m² and the maximal installable capacity is 26 000 m². We can modify the capacity of a facility by adding capacity options (such as surface extension or new production chain). For each facility, we determine a set of available capacity options. Another assumption is that a facility will never be run at 1% or at 100% of its capacity. To avoid these scenarios, we introduce two new parameters: minimal and maximal rates of utilisation for each facility. Moreover, a facility cannot reach its maximal rate of utilisation in the opening period because time is needed to reach full capacity (installing machines; hiring and training employees). In our formulation, this parameter increases with time.

Each facility has a general common capacity for all manufactured products. To simplify the calculation of the necessary capacity, the workload is calculated in a generic unit and the capacity of a facility is expressed in the same unit.

The production process can be divided into several steps and can be shared between several plants. Each facility/warehouse can manufacture/store a subset of products. We can also resort to subcontracting for the manufacturing of some intermediate components. We assume that subcontracted products are manufactured in an external plant of our network and are delivered directly to the plant. The capacity of the external plant is supposed to be infinite.

We suppose that the inventories are held only in warehouses, not in plants. One point that distinguishes plants from warehouses is that plants have only a production capacity, also called a treatment capacity, while warehouses have two different capacities: treatment and storage. We introduce seasonal inventories to allow the smoothing of product over the planning horizon. On the other hand, we define the number of deliveries from plants to warehouses in one period in order to deduce the average quantity of each shipment, which is used to calculate the cyclic inventory.

We make a distinction between a private warehouse (owned by the company) and a public warehouse (hired by the company). The status of a public warehouse can change more than once during the planning horizon.

For modelling purposes, the cost structure of the objective function consists of two parts: fixed costs

and variable costs. Public warehouses have no fixed costs but their variable costs are higher than those of private warehouses. All our costs are linear and independent of time.

2.2. Position of the proposed model

We propose a new dynamic model for facility location in supply chain management. Our model is similar to two previous works, which are the static model proposed by Paquet et al. (2004) and the dynamic model proposed by Martel (2005). All these models are multi-echelon network with capacitated external suppliers and deterministic customer demands. The modular capacity and the bill of materials are also taken into consideration. Regarding facility layout, each facility can treat only a subset of manufactured products. Concerning system cost structure, all three models propose an objective function with two separate parts: fixed costs and variable costs.

The main difference between our model and that proposed by Paquet et al. (2004) is that ours is dynamic. We consider a planning horizon of several periods so that the model has to determine the best period for opening or closing facilities. Another decision linked to the dynamic model is inventory management, which is also taken into consideration in our model. Nevertheless, Paquet et al. (2004) consider different transportation modes while we suppose only one in our formulation.

Another similar work is the model proposed by Martel (2005). The author considers an international network with a complex calculation of the after-tax profit that we do not consider. However, our contribution compared to Martel's model is the consideration of two types of warehouse (private and public) corresponding to the possibility of subcontracting. On the other hand, we define two limit levels (maximal and minimal) of utilisation for each facility. Additionally, the maximal rate of utilisation increases with time from the moment of opening. It makes the formulation more realistic and enables the transition phase after opening to be modelled. In the same way, we can decrease the rate of utilisation before closing a facility. The possibility of obtaining a discount by ordering several raw materials from the same supplier is also a new point. This assumption adds a new criterion for supplier selection.

2.3. The model

2.3.1. Data sets and variables

$\mathcal{T} (t \in \mathcal{T})$ planning horizon
 $\mathcal{S} (s \in \mathcal{S})$ set of suppliers
 $\mathcal{I} (i \in \mathcal{I})$ set of plants
 \mathcal{I}_c set of plants that may close ($\mathcal{I}_c \subset \mathcal{I}$)
 \mathcal{I}_o set of plants that may open ($\mathcal{I}_o \subset \mathcal{I}$)
 $\mathcal{J} (j \in \mathcal{J})$ set of warehouses
 \mathcal{JP} set of private (permanent) warehouses
 \mathcal{JP}_c private warehouses that may close ($\mathcal{JP}_c \subset \mathcal{J}$)
 \mathcal{JP}_o private warehouses that may open ($\mathcal{JP}_o \subset \mathcal{J}$)
 \mathcal{JH} set of public (hired) warehouses
 \mathcal{JH}_c public warehouses that may stop being hired ($\mathcal{JH}_c \subset \mathcal{J}$)
 \mathcal{JH}_o public warehouses that may begin being hired ($\mathcal{JH}_o \subset \mathcal{J}$)
 $\mathcal{O} (o \in \mathcal{O})$ set of capacity options
 $\mathcal{C} (c \in \mathcal{C})$ set of customers
 $\mathcal{P} (p \in \mathcal{P})$ set of products
 \mathcal{P}_r set of raw materials ($\mathcal{P}_r \subset \mathcal{P}$)
 \mathcal{P}_m set of manufactured products ($\mathcal{P}_m \subset \mathcal{P}$)
 \mathcal{P}_f set of finished products ($\mathcal{P}_f \subset \mathcal{P}$)
 Γ_i^+, Γ_i^- set of successors and predecessors of a node i

2.3.1.1. Suppliers

$HD_{s,p}^t$ 1 if the raw material p is available from s at t ; 0 otherwise
 $\bar{A}_{s,p}^t$ available capacity of supplier s for p at t

2.3.1.2. Plants

\underline{K}_i initial production capacity at i
 \bar{K}_i maximal installable production capacity at i
 $HA_{i,o}^t$ 1 if we can add the option o to facility i at t ; 0 otherwise
 KT_o capacity of option o
 \underline{U}_i minimal percentage of utilisation of facility i (independent of t)
 \bar{U}_i^t maximal percentage of utilisation of facility i at t
 $HP_{i,p}^t$ 1 if the plant i can manufacture p at t ; 0 otherwise

2.3.1.3. Warehouses. The warehouses have almost the same characteristics as the plants. The only distinctions are the following:

\underline{S}_j initial storage capacity at j
 KS_o storage capacity of option o (if o is added to a warehouse)
 $HS_{j,p}^t$ 1 if the warehouse j can store p at t ; 0 otherwise

2.3.1.4. Customers

$D_{c,p}^t$ demand of customer c for product p at t .

2.3.1.5. Products

HC_p 1 if the product p can be subcontracted
 $B_{p',p}$ quantity of p' necessary to manufacture a unit of p (bill of materials)
 $LT_{p,i}$ workload for the treatment of a unit p at facility i ($i \in \mathcal{I} \cup \mathcal{J}$)
 $LS_{p,j}$ workload for the storage of a unit p at warehouse j ($j \in \mathcal{J}$),
 $A_{i,j}$ number of deliveries from plant i to warehouse j in one period

2.3.1.6. Costs

CF_s fixed cost for selecting a supplier
 CD_s discount for at least two types of raw materials ordered at supplier s
 CO_i fixed cost for opening a facility at a potential location i
 CC_i fixed cost for closing an existing facility i
 $CA_{i,o}$ fixed cost for adding capacity option o to a facility i
 $CU_{i,0}$ fixed cost for operating a facility i
 $CU_{i,o}$ fixed cost for operating capacity option o at facility i
 $CP_{p,i}$ treatment cost of a unit p at facility i
 $CP_{p,0}$ subcontracting cost of a unit of p
 $CS_{p,j}$ storage cost of a unit of p at warehouse j
 $CT_{p,i,j}$ transportation cost of a unit of p from i to j

2.3.1.7. Decision variables. The model considers four binary variables:

x_i^t 1 if the entity i is active at t ; 0 otherwise
 $y_{i,o}^t$ 1 if the capacity option o is added to i at t ; 0 otherwise
 $z_{s,p}^t$ 1 if the supplier s is selected for the raw material p at t ; 0 otherwise
 v_s^t 1 if the supplier s gives a discount at t ; 0 otherwise

We have the same variable x for suppliers, plants and warehouses knowing that a supplier is active if it delivers at least one raw material and a facility/warehouse is active if it is opened in the corresponding period.

2.3.1.8. Other variables are continuous

- $f_{p,i,j}^t$ quantity of product p transferred from location i to j at t
- $k_{p,i}^t$ quantity of product p subcontracted for plant i at t
- $g_{p,i}^t$ quantity of product p produced in plant i at t
- $h_{p,j}^t$ quantity of product p held in warehouse j at the beginning of t

2.3.2. The constraints

In this section, we present the constraints of our model. We divide them into four categories: demand satisfaction constraints; capacity constraints; logical constraints; integrity and non-negativity constraints.¹

2.3.2.1. Demand satisfaction constraints. The constraints (1) guarantee that all customer demands are met, for all products required and for any period. The constraints (2) are the flow conservation at warehouses. They must receive enough finished products from plants in order to meet all the demands (in consideration of seasonal stock). The constraints (3) ensure that plants receive enough raw materials and semi-finished products in order to manufacture the required quantity of finished products. The last constraints (4) require that the quantity of manufactured and subcontracted products at a plant is greater than or equal to its delivered quantity.

$$\sum_{j \in \Gamma_c^-} f_{p,j,c}^t \geq D_{c,p}^t \quad \forall c \in \mathcal{C}, \quad \forall p \in \mathcal{P}_f, \quad (1)$$

$$h_{p,j}^{t-1} + \sum_{i \in \Gamma_j^-} f_{p,i,j}^t = \sum_{c \in \Gamma_j^+} f_{p,j,c}^t + h_{p,j}^t \quad \forall j \in \mathcal{J}, \quad \forall p \in \mathcal{P}_f, \quad (2)$$

$$\sum_{i \in \Gamma_i^-} f_{p',i,i}^t + k_{p',i}^t \geq \sum_{p \in \mathcal{P}_m} B_{p',p} \cdot g_{p,i}^t \quad \forall i \in \mathcal{I}, \quad \forall p' \in \mathcal{P}_r \cup \mathcal{P}_m, \quad (3)$$

$$g_{p,i}^t + k_{p,i}^t \geq \sum_{j \in \Gamma_i^+} f_{p,i,j}^t \quad \forall i \in \mathcal{I}, \quad \forall p \in \mathcal{P}_m. \quad (4)$$

2.3.2.2. Capacity constraints. Plants and warehouses have an initial capacity. The installed capacity is defined as the sum of the initial capacity and the capacity of the added options. We suppose that a facility cannot function under its minimum rate of utilisation and cannot exceed the maximum rate of utilisation of its installed capacity. The two constraints (5) and (6) represent these conditions for the plants. The constraints (7) and (8) are similar but applied to the warehouses. Moreover, warehouses must not store more than their storage capacity (9). Finally, the installed capacity at any facility must not exceed its maximal installable capacity (10).

$$\sum_{p \in \mathcal{P}_m} LT_{p,i} \cdot g_{p,i}^t \leq \bar{U}_i \cdot \left(\underline{K}_i \cdot x_i^t + \sum_{o \in \mathcal{O}} KT_o \cdot y_{i,o}^t \right) \quad \forall i \in \mathcal{I}, \quad (5)$$

$$\sum_{p \in \mathcal{P}_m} LT_{p,i} \cdot g_{p,i}^t \geq \underline{U}_i \cdot \left(\underline{K}_i \cdot x_i^t + \sum_{o \in \mathcal{O}} KT_o \cdot y_{i,o}^t \right) \quad \forall i \in \mathcal{I}, \quad (6)$$

$$\sum_{p \in \mathcal{P}_f} \sum_{c \in \Gamma_j^+} LT_{p,j} \cdot f_{p,j,c}^t \leq \bar{U}_j \cdot \left(\underline{K}_j \cdot x_j^t + \sum_{o \in \mathcal{O}} KT_o \cdot y_{j,o}^t \right) \quad \forall j \in \mathcal{J}, \quad (7)$$

$$\sum_{p \in \mathcal{P}_f} \sum_{c \in \Gamma_j^+} LT_{p,j} \cdot f_{p,j,c}^t \geq \underline{U}_j \cdot \left(\underline{K}_j \cdot x_j^t + \sum_{o \in \mathcal{O}} KT_o \cdot y_{j,o}^t \right) \quad \forall j \in \mathcal{J}, \quad (8)$$

$$\sum_{p \in \mathcal{P}_f} LS_{p,j} \cdot \left(h_{p,j}^t + \sum_{i \in \Gamma_j^-} \frac{1}{\Lambda_{ij}} \cdot f_{p,i,j}^t \right) \leq \underline{S}_j \cdot x_j^t + \sum_{o \in \mathcal{O}} KS_o \cdot y_{j,o}^t \quad \forall j \in \mathcal{J}, \quad (9)$$

$$\underline{K}_i \cdot x_i^t + \sum_{o \in \mathcal{O}} KT_o \cdot y_{i,o}^t \leq \bar{K}_i \quad \forall i \in \mathcal{I} \cup \mathcal{J}. \quad (10)$$

2.3.2.3. Logical constraints. The constraints (11) ensure that suppliers deliver a raw material if and only if they are selected for this raw material. Moreover, their delivery cannot exceed their capacity. The constraints (12) state that a supplier,

¹In order to simplify the notations, the expression $\forall t \in \mathcal{T}$ will be omitted in every constraint.

whenever selected, can only deliver a subset of raw materials. We suppose that a supplier offers a discount if it is selected for at least two raw materials (13). The constraints (14) propose a minimal amount of the delivered quantity of each supplier in order to avoid small artificial orders for obtaining the discount. This limit, α , is arbitrary and has to be adjusted according to different situations. The constraints (15) and (16) state that a plant/warehouse can only deliver products that it can manufacture/store. The constraints (17) require that only opened plants can subcontract products. M is a large arbitrary number. The constraints (18) ensure that an opened facility can add available capacity options only.

$$\sum_{i \in \Gamma_s^+} f_{p,s,i}^t \leq z_{s,p}^t \cdot \bar{A}_{s,p}^t \quad \forall s \in \mathcal{S}, \forall p \in \mathcal{P}_r, \quad (11)$$

$$z_{s,p}^t \leq x_s^t \cdot HD_{s,p}^t \quad \forall s \in \mathcal{S}, \forall p \in \mathcal{P}_r, \quad (12)$$

$$v_s^t \leq \left(1/2 \sum_{p \in \mathcal{P}_r} z_{s,p}^t \right) \quad \forall s \in \mathcal{S}, \quad (13)$$

$$\sum_{i \in \Gamma_s^+} f_{p,s,i}^t \geq \alpha \cdot \bar{A}_{s,p}^t \cdot z_{s,p}^t \quad \forall s \in \mathcal{S}, \forall p \in \mathcal{P}_r, \quad (14)$$

$$f_{p,i,i'}^t \leq HP_{i,p}^t \cdot \bar{K}_i \quad \forall i \in \mathcal{I}, \forall i' \in \Gamma_i^+, \forall p \in \mathcal{P}_m, \quad (15)$$

$$f_{p,j,c}^t \leq HS_{j,p}^t \cdot \bar{K}_j \quad \forall j \in \mathcal{J}, \forall c \in \Gamma_j^+, \forall p \in \mathcal{P}_f, \quad (16)$$

$$k_{p,i}^t \leq x_i^t \cdot HCp \cdot M \quad \forall i \in \mathcal{I}, \forall p \in \mathcal{P}_m, \quad (17)$$

$$y_{i,o}^t \leq x_i^t \cdot HA_{i,o}^t \quad \forall i \in \mathcal{I} \cup \mathcal{J} \cup \mathcal{P}, \forall o \in \mathcal{O}. \quad (18)$$

On the other hand, the constraints (19) and (20) prevent the facilities from changing their status (open or closed) more than once. Closed facilities cannot be reopened, and new facilities will remain in activity until the end of the planning horizon. For the public warehouses, we suppose that their status can be changed after at least two periods. The constraints (21) and (22) establish this condition. Additionally, we can add new capacity options but we cannot remove them (23) and a facility where capacity options are added cannot be closed in the subsequent periods (24).

$$x_i^t \geq x_i^{t+1} \quad \forall i \in \mathcal{I}_c \cup \mathcal{J} \cup \mathcal{P}_c, \quad (19)$$

$$x_i^t \leq x_i^{t+1} \quad \forall i \in \mathcal{I}_o \cup \mathcal{J} \cup \mathcal{P}_o, \quad (20)$$

$$2 \cdot x_j^t - x_j^{t-1} - x_j^{t+1} \geq -1 \quad \forall j \in \mathcal{J} \cup \mathcal{H}, \quad (21)$$

$$2 \cdot x_j^t - x_j^{t-1} - x_j^{t+1} \leq 1 \quad \forall j \in \mathcal{J} \cup \mathcal{H}, \quad (22)$$

$$y_{i,o}^t \leq y_{i,o}^{t+1} \quad \forall i \in \mathcal{I} \cup \mathcal{J} \cup \mathcal{P}, \quad o \in \mathcal{O}, \quad (23)$$

$$x_i^t \geq y_{i,o}^{t-1} \quad \forall i \in \mathcal{I} \cup \mathcal{J} \cup \mathcal{P}, \quad o \in \mathcal{O}. \quad (24)$$

2.3.2.4. *Integrity and non-negativity constraints.* This set of constraints focuses on the nature of variables. The constraints (25)–(28) require that these four variables are binary. The constraints (29)–(32) restrict all other variables from taking non-negative values.

$$x_i^t \in \{0, 1\} \quad \forall i \in \mathcal{S} \cup \mathcal{I} \cup \mathcal{J}, \quad (25)$$

$$y_{i,o}^t \in \{0, 1\} \quad \forall i \in \mathcal{I} \cup \mathcal{J}, \quad \forall o \in \mathcal{O}, \quad (26)$$

$$z_{s,p}^t \in \{0, 1\} \quad \forall s \in \mathcal{S}, \quad \forall p \in \mathcal{P}_r, \quad (27)$$

$$v_s^t \in \{0, 1\} \quad \forall s \in \mathcal{S}, \quad (28)$$

$$f_{p,i,i'}^t \geq 0 \quad \forall i \in \mathcal{S} \cup \mathcal{I} \cup \mathcal{J}, \quad \forall i' \in \mathcal{I} \cup \mathcal{J} \cup \mathcal{C}, \quad p \in \mathcal{P}, \quad (29)$$

$$k_{p,i}^t \geq 0 \quad \forall i \in \mathcal{I}, \quad p \in \mathcal{P}_m, \quad (30)$$

$$g_{p,i}^t \geq 0 \quad \forall i \in \mathcal{I}, \quad p \in \mathcal{P}_m, \quad (31)$$

$$h_{p,j}^t \geq 0 \quad \forall j \in \mathcal{J}, \quad p \in \mathcal{P}_f. \quad (32)$$

2.3.3. *The valid inequalities*

To strengthen the linear programming relaxation and to have a better lower bound, we can add some valid inequalities to the model.

$$\sum_s \bar{A}_{s,p'}^t \geq \sum_{p \in \mathcal{P}_m} B_{p',p} \cdot \left(\sum_{i \in \mathcal{I}} g_{p,i}^t \right) \quad \forall p' \in \mathcal{P}_r. \quad (33)$$

These inequalities (33) show that the total available capacity of all suppliers must be greater than the demands of plants.

$$\sum_{j \in \mathcal{J}} \left(\bar{K}_j \cdot x_j^t + \sum_{o \in \mathcal{O}} KT_{o,j} \cdot y_{j,o}^t \right) \geq \sum_{c \in \mathcal{C}} \sum_{p \in \mathcal{P}_f} D_{c,p}^t. \quad (34)$$

Assuming that the workload at each warehouse is at least one unit, the sum of maximal installable capacities of warehouses is greater than the sum of customer demands (34). We cannot define a similar inequality for the plants because we have the possibility of subcontracting and the capacity of external plants is supposed to be infinite.

These valid inequalities have been introduced by [Jardin et al. \(2006\)](#). Their effect is to decrease the computational time by a factor of 1.2 to 2.

2.3.4. *The objective function*

The objective function is to minimise the sum of the different following components:

$$\begin{aligned} \text{Min } f(x, y, z, v) = & \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} CF_{s,i} \cdot x_s^t - CD_{i,o} \cdot v_o^t \\ & \text{(supplier selection fixed cost)} \end{aligned} \quad (35)$$

$$+ \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}_o \cup \mathcal{J} P_o} CO_i \cdot (x_i^{t+1} - x_i^t) \tag{36}$$

(opening facility fixed cost)

$$+ \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}_c \cup \mathcal{J} P_c} CC_i \cdot (x_i^t - x_i^{t+1}) \tag{37}$$

(closing facility fixed cost)

$$+ \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I} \cup \mathcal{J} \mathcal{P}} \sum_{o \in \mathcal{O}} CA_{i,o} \cdot (y_{i,o}^{t+1} - y_{i,o}^t) \tag{38}$$

(adding capacity options cost)

$$+ \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I} \cup \mathcal{J} \mathcal{P}} CU_{i,0} \cdot x_i^t + \sum_{o \in \mathcal{O}} CU_{i,o} \cdot y_{i,o}^t \tag{39}$$

(operating facility fixed cost)

$$+ \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}_m} \sum_{i \in \mathcal{I}} CP_{p,i} \cdot g_{p,i}^t + CP_{p,0} \cdot k_{p,i}^t \tag{40}$$

(production variable cost)

$$+ \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}_f} \sum_{j \in \mathcal{J}} CS_{p,j} \cdot \left(h_{p,j}^t + \sum_{i \in \Gamma_j^-} \frac{f_{p,i,j}^t}{2A_{i,j}} \right) \tag{41}$$

(storage variable cost)

$$+ \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}_f} \sum_{j \in \mathcal{J}} \sum_{c \in \Gamma_j^+} CP_{p,j} \cdot f_{p,j,c}^t \tag{42}$$

(distribution variable costs)

$$+ \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{I} \cup \mathcal{J} \cup \mathcal{J} \mathcal{P}} \sum_{j \in \Gamma_i^+} CT_{p,i,j} \cdot f_{p,i,j}^t \tag{43}$$

(transportation variable cost).

The first five components are fixed costs. Concerning the supplier selection fixed costs (35), the company has to pay them for each period and for each selected supplier. Whenever possible, the discount CD_s is removed from the cost. Other costs that are paid once a year are operating fixed costs (39). Other fixed costs are paid just once during the whole planning horizon, like the opening/closing costs and adding capacity options (36)–(39). Indeed, these costs are considered as an investment for a new site (opening and adding capacity option costs) or as a cost for removing installed things and for paying for dismissing people.

The second part of the objective function consists of variable costs that depend on the product flows in the network. The production costs (40) are the sum of the product costs for internal plants and the subcontracting costs. Storage variable costs (41) are

calculated as a function of the average inventory level. In our model, the average inventory level is defined as half of the delivered quantity (from plants) which is calculated by the quotient of the total transported quantity divided by the number of deliveries in one period. The next cost is the distribution one (42), which is calculated as a function of the delivered quantity from warehouses to customers. Finally, the transportation costs (43) are related to the product flows between each node in the network.

3. Computational tests

In order to evaluate our model, we apply it to a realistic scenario in which the customer demands increase. Data sets with ten different sizes are described in Section 3.1. A summary of the computational results is presented in Section 3.2. For more details about the data generator, interested readers should refer to Appendix A.

3.1. Description of the data

The difficulty of the problem is mainly determined by three criteria: the size of the network, the structure of the supply chain and the bill of materials. The first element is given by the number of suppliers ($|S|$), plants ($|I|$), warehouses ($|J|$) and customers ($|C|$). In our data sets, $|S|$, $|I|$ and $|J|$ are stretched in the interval $[5, 15]$ and $|C|$ varies in the interval $[50, 160]$.

The structure of the supply chain is defined by the possibility of getting raw materials from each supplier, from the possibility of adding capacity options and by the manufacturing or storing possibilities of each facility. These possibilities determine the density of the network.

The first indication of the bill of materials is the total number of products. In our tests, this value lies in the interval $[10, 26]$. We impose that semi-finished products are manufactured from raw materials and finished products are manufactured from raw materials and semi-finished products.

3.2. Numerical results

The problem instances are divided into two sets: five families of small instances (denoted in increasing order from $S1$ to $S5$) and five families of larger instances (denoted in increasing order from $L1$ to $L5$). In the small instances, the number of nodes

varies from 65 to 120 and there are between 10 and 18 products. For larger instance, the number of nodes varies in the interval [144, 205] and there are between 18 and 26 products. The planning horizon is fixed at five periods. In each family, we generate 10 instances with the same characteristics. This operation limits the impact of particular data sets.

These instances are solved by Xpress-MP (release 2005) on a Pentium IV, 3.2 GHz processor with 1 GHz of RAM. Xpress-MP is an MILP solver based on the branch-and-bound approach. At each step, several strategies are proposed to construct the candidate set. The usual default strategy is local first, i.e. two descendants of the current nodes form the candidate set, but if both have been fathomed then all active nodes form the candidate set. We report here the number of generated nodes in the branch-and-bound tree. The displayed values are summarised from the average of 10 instances of each family. We report also the total number of binary variables; however, only half of them are location variables (variables x and y). The others concern supplier management (variables z and v).

For small instances, we report in Table 3 their CPU times to reach the optimal solution.

We observe that the longest CPU time on these instances is about 3 h. Thus, we limit the computational time to 3 h for every larger instance. The optimality gaps obtained within this time are represented in Table 4. The formulation to calculate optimality gap is

$$\text{Gap} = \frac{\text{Best solution} - \text{Best lower bound}}{\text{Best solution}} \times 100.$$

When the size of instances gets larger, we notice an increase in all observed values: minimum, average and maximum of the CPU times; minimum, average and maximum of the optimality gaps. Additionally, we point out that for the larger instances, within 3 h, the obtained optimality gaps are reasonable. Most of them are less than 1%. Concerning branch-and-bound generated nodes, we remark that for small families, when instances get larger, the total number of generated nodes increases. This means that for larger problems, branch-and-bound needs to generate more nodes to find an optimal solution. However, for larger instance families, we notice that the average generated nodes for $L5$ is less than that for $L1$. This opposite evolution is due to the limited execution time for these instances. For a larger instance, branch-and-bound needs more time to solve a node. Hence, for the same limited time (3 h), branch-and-bound generates fewer

Table 3
Computational results for small instances

Parameters					
Example	S1	S2	S3	S4	S5
Suppliers	5	10	10	12	12
Plants	5	9	9	9	9
Warehouses	5	9	9	9	9
Customers	50	70	70	90	90
Products	10	10	14	14	18
Binary variables	375	710	810	900	1020
<i>CPU times (s)</i>					
Min.	5	38	237	274	748
Average	24	144	1099	4500	6840
Max.	53	638	1954	10 920	11 100
<i>Branch-and-bound generated nodes</i>					
Min.	41	107	323	785	1318
Average	327	517	3336	13 193	17 826
Max.	951	1562	11 851	60 628	67 067

Table 4
Computational results for larger instances within 3 h

Parameters					
Example	L1	L2	L3	L4	L5
Suppliers	12	12	14	14	15
Plants	11	11	11	11	15
Warehouses	11	11	11	11	15
Customers	110	110	130	130	160
Products	18	22	22	26	26
Binary variables	1100	1220	1350	1490	1725
<i>Optimality gap (%)</i>					
Min.	0	0	0	0.6	1.6
Average	0.5	0.7	0.8	1	2
Max.	1	1	2	2	3.8
<i>Branch-and-bound generated nodes</i>					
Min.	3333	2062	1090	724	121
Average	4560	3457	3207	2121	360
Max.	7216	6242	6611	3273	976

nodes. This explanation is also valid for the increase in the optimality gaps.

On the other hand, we observe a strong variability in the computational times for instances of the same family. This can be explained by the nature of our data sets. Three criteria influence the difficulty of the problem: the size of the network, the structure of the supply chain and the bill of materials. For instances of the same family, we have the same size of network but we generate instances with different levels of difficulty for the two other criteria.

One instance of the family S1 is detailed for illustration purposes (Fig. 2). We consider a network where two suppliers out of five are selected, two plants are operating and three may be open at the initial period. As regards warehouses, two private ones are already operational and one public warehouse can be hired at any time. There are also two available sites to build new private warehouses.

Fig. 2 depicts the network evolution during five periods. Every active supplier is filled in black. For every facility, the black part represents the rate of capacity utilisation. The first warehouse is public. Since the customers are the same for every period, we do not represent them on the figure. Product flows along the supply chain are also omitted in order to keep the figure clear and to focus on the facility location aspect.

We observe that the main changes in location are made in the second period. A new plant is opened and two capacity options are implemented for the

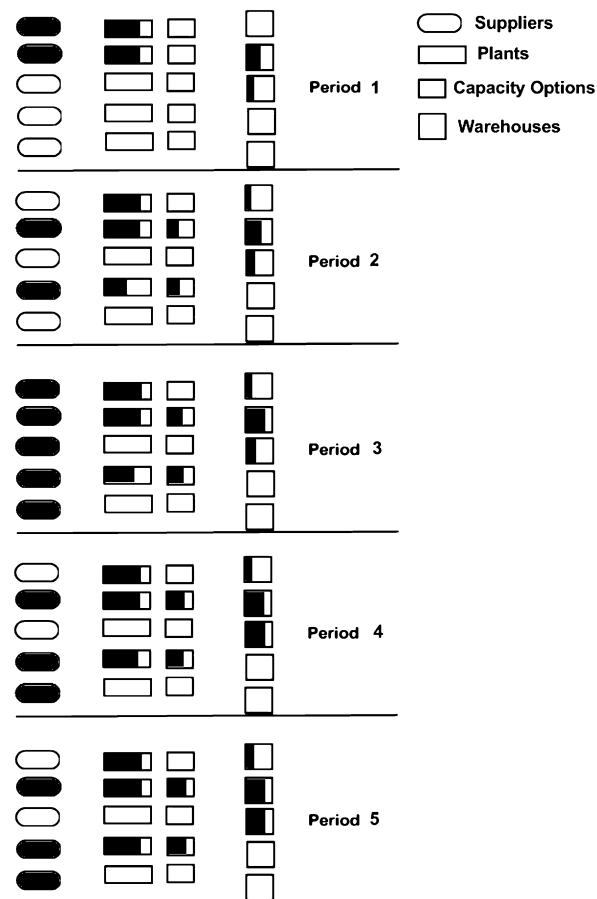


Fig. 2. Evolution of the network.

facilities 2 and 4. Option capacity is added while 100% of plant capacity is not reached because of the constraints that forbid the use of 100% of the capacity. Additionally, we can point out the increase in the rate of utilisation of plant 4 and of two capacity options. We can consider it as a transition phase after the opening. Regarding warehouse management, it is preferable to use the existing warehouses as much as possible and to hire public warehouses when the private ones are saturated.

The investment in the second period explains the peak of the total cost in Fig. 3. The cost growth in the last two periods can be explained by the increase in customer demands, leading to an increase in the activity. On the other hand, we can also remark that after five periods, the highest cost is the operating facilities fixed cost, followed by the production variable cost and the transportation cost while the supplier selection cost is negligible compared to the others.

In order to meet all customer demands, three modes of procurement are available: manufactured products, subcontracted products and stored products. To have a more detailed view of the production management, Table 5 describes the origin of the delivered products in each period. Manufactured products are represented in the column *Internal*, subcontracted products are represented in the second column and the third column is the rate of products that are directly taken from the inventory. We have five manufactured products: two semi-finished products (Products 1 and 2) and three finished products (Products 3, 4 and 5).

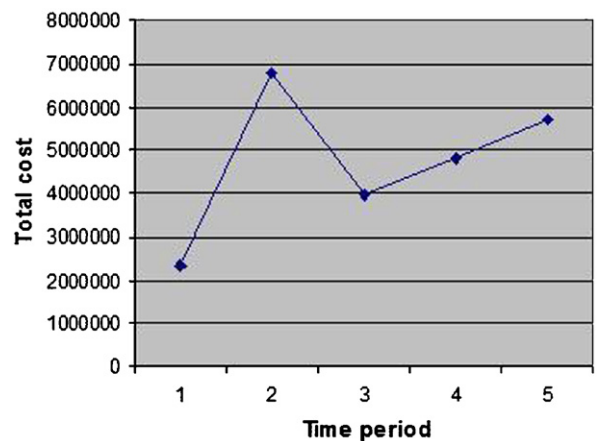


Fig. 3. Total cost per time period.

Table 5
Production activities

T	Product 1		Product 2		Product 3		Product 4		Product 5	
	Internal	Subcontracted	Internal	Subcontracted	Internal	Subcontracted	Internal	Subcontracted	Internal	Subcontracted
1	0	0	0.25	0.75	1	0	0	1	0	0
2	0	0	0	1	1	0	0	1	0.89	0
3	0	0	1	0	0.45	0	0	0	0.72	0
4	0	0	0.64	0.34	0.77	0	0	1	0.63	0
5	0	0	0	1	0.95	0	0	1	0.57	0
		From inventory		From inventory		From inventory		From inventory		From inventory
		0		0		0		0		0
		0		0		0		0		0.11
		0		0		0.55		0		0.28
		0		0		0.23		0		0.37
		0		0		0.05		0		0.43

We remark that the model proposes different production policies for the five manufactured products. Products 1, 3 and 5 are entirely manufactured by the internal plants while product 4 is fully subcontracted. Production management for product 2 is the most complicated. The changes can be explained as follows. For the first two periods, the lack of plant capacity leads to subcontracting. Even after a new plant is opened, it is preferable to use its capacity to manufacture products 3 and 5 in order to store them for the next period. In the third period, the rate of utilisation of the new plant and new capacity options increase. Moreover, the system requires fewer products 3 and 5 thanks to stored products. This is why the production of product 2 becomes internal. This scenario appears to be unrealistic and shows the limits of any generic model. For practical uses, we could customise the model by appending new constraints and rules given by each company.

4. Conclusion and further research

We have proposed a new dynamic model for facility location and supply chain design. Our approach can be used to design a new network, to re-engineer an existing network or to evaluate a strategic decision. The main contribution of the present work is to extend the existing models found in the literature, with the goal of complying with the complexity of industrial cases. We integrate original characteristics such as the existence of capacity options, transition phases after opening a facility, the distinction between public and private warehouses, and a possible discount offered by suppliers.

Numerical experiments for solving our model with a standard solver illustrate that it is possible to obtain optimal or near optimal solutions within a reasonable amount of time, for small and medium-size instances. However, for the largest instances tested in this paper, achieving to the optimal solution would require much more than 3 h. Hence, this seems to reach the limit of such a solution technique. For the next step of our research, we would like to develop some heuristics and meta-heuristics for this specific family of problems as well as exact decomposition methods.

The model could still be refined with several extensions, such as adding some budget constraints in order to address financial considerations or adding more realistic cost functions for each manufacturing possibility. We would, for example,

be able to model investments in new technologies adequately. Another direction for improvement could be to include tactical considerations in order to take into account the seasonality of demand.

Appendix A. Description of the data generator

In this appendix, we describe the rules that we used to generate the 100 data sets mentioned in Section 3. For each instance test, the data generator has to produce:

- Customer demands: for each finished product and for each time period.
- Bill of material: quantity of raw materials and semi-finished products needed for each manufactured product.
- Possibility of getting a raw materials from each supplier.
- Possibility of manufacturing a product at each plant.
- Possibility of treating and storing a finished product at each warehouse.
- Possibility of adding a new capacity option to each facility.
- Workload for the treatment/storage of a product at each facility/warehouse.
- Initial and maximal capacity for each facility—capacity for each capacity option.
- Evolution of the maximal rate of utilisation for each facility and capacity option.
- Fixed costs: supplier selection cost, discount from a supplier, opening cost, closing cost, operating cost, adding capacity option cost.
- Variable costs: treatment cost, subcontracting cost, storage cost and transportation cost.

A.1. Customer demands

We adopt the same procedure as Melo et al. (2006). For the first period, the customer demands are randomly generated according to a uniform distribution in one predefined interval. In our test, we suppose that we have 90% of standard customers and 10% of important customers. Standard customer demands are generated in the interval [80, 130] and important customer demands are generated in the interval [150, 300]. For the other periods, we take the demands of the preceding period as a reference and apply a multiplicative factor $(1 + \varepsilon)$ where ε is randomly chosen in the interval [0, 1].

A.2. Bill of materials

We impose that semi-finished products are manufactured from raw materials and finished products are manufactured from raw materials and semi-finished products. The necessary quantity of each component is randomly selected in the set $\{0, 3\}$.

At the end of the procedure, we check that each raw material and each semi-finished product is used by at least one manufactured product, and that each manufactured product has at least one component.

A.3. Possibility parameters

This category defines the possibility of getting a raw material from each supplier; the possibility of manufacturing a product at each plant; the possibility of treating and storing a finished product at each warehouse; the possibility of adding a new capacity option to each facility.

These data are generated with the same procedure proposed by Cordeau et al. (2006). The following example illustrates how we generate these data. We consider a raw material and determine first a set of suppliers that can provide this raw material, called S^* . The cardinal of S^* is chosen randomly in the interval $[2, |S|]$. The corresponding number of suppliers is then selected randomly from the set of S compatible suppliers. The same procedure is used for other data in this category.

A.4. Workload for the treatment and for the storage of each product

We suppose that the workload for the treatment is approximately twice that for the storage of the same product. Consequently, workloads for the treatment are randomly generated with a uniform distribution in the interval [1, 5] while the workloads for storage are generated in the interval [1, 3].

A.5. Initial and maximal facility capacity generator

We distinguish the procedure to generate capacity for opened facilities and for potential facilities. As regards opened facilities, for the first period, we suppose that all location decisions have already been taken. Thus, opened facility capacities must be sufficient to meet the customer demands in the first period (in order to guarantee the feasibility of the solution).

The relation between opened facility capacities and customer demands is inspired by Melkote and Daskin (2001). We calculate the total customer demands in the first period for all finished products. Then we multiply this sum by the maximal treatment workload. This total charge must be ensured by opened facilities. Then, we define the facility capacity by dividing the total charge by the number of opened facilities.

Regarding a potential facility, as we consider the scenario where the customer demands increase, then we suppose that when the company decides to open a new facility, it must be larger than the old one. That is why, for potential capacity, we generate randomly a number in the interval $[0, 2]$ which is the capacity multiple indicator with reference to the opened facility capacity.

Concerning maximal capacity, for every facility, we restrict its maximal capacity to 1.5 times its initial one.

We must also define the capacity of each option. For these instance tests, we suppose that each facility can add no more than three capacity options whose capacities are, respectively, $\frac{1}{3}$, $\frac{1}{6}$ and $\frac{1}{12}$ of the initial capacity of the opened facility.

A.6. Evolution of the maximal rate of utilisation

We take into consideration the assumption that a facility cannot reach its maximal capacity after its opening. Then we introduce a new parameter that represents the maximal rate of utilisation of each facility in each period. This parameter increases with time. The rule to define the evolution of this parameter is inspired by Melachrinoudis and Min (2000). The authors consider a real case study of a company manufacturing and selling chain link fences and their related hardware items. This American enterprise would like to move its current manufacturing and warehousing facility to a new state. They estimate that their facility can reach full capacity only after 2 yr due to the transition phase. We adopt the same value for our maximal rate of utilisation: 50% for the first year after the opening, 70% for the second year and 90% after 2 yr.

The same argument can be used to model the transition phase before the closing of a facility.

A.7. Fixed cost generator

Almost all fixed costs are generated with respect to facility capacity.

The following formula is presented in the paper of Cortinhal and Captivo (2003). The authors define the fixed cost for location of a facility, noted f_i , as

$$f_i = U[0, 90] + U[100, 110] \times \sqrt{a_i},$$

where $U[x, y]$ represents a uniform distribution in the range $[x, y]$ and a_i is the facility capacity. For our instance tests, we adopt the same formula to calculate the opening fixed costs.

Other fixed costs are generated with the same formula except for the range of the uniform distribution:

- Opening fixed costs: $U[0, 90] + U[100, 110] \times \sqrt{a_i}$.
- Closing fixed costs: $U[0, 50] + U[20, 30] \times \sqrt{a_i}$.
- Operating fixed costs: $U[0, 20] + U[10, 20] \times \sqrt{a_i}$.
- Adding capacity option: $U[0, 30] + U[30, 40] \times \sqrt{a_i}$.
- Supplier selection fixed costs: $U[100, 200]$.
- Discount from supplier: 20% of the supplier selection fixed costs.

We suppose that the closing cost is cheaper than the opening one. On the other hand, when a facility is opened, the adding capacity option cost is lower than the cost at the beginning. As regards the supplier selection cost, we estimate this to be negligible in comparison with other costs. The main criteria that influence the supplier selection are their availability and the transportation costs from suppliers to plants.

A.8. Variable costs

The relation between the treatment costs of plants and the treatments costs of warehouses is inspired by Melachrinoudis and Min (2000). In their real-life example, the authors estimate that the unitary production cost varies from 10.79\$ to 16.20\$ while the unitary storage cost varies from 2.00\$ to 4.00\$. In our data generator, the production costs are generated randomly with a uniform distribution in the interval $[10, 20]$ and the storage costs in the interval $[2, 5]$.

Regarding subcontracting costs and storage costs in hired warehouses, we estimate that these costs are twice the standard production cost of the internal production cost and storage cost in private warehouses. We suppose that the transportation costs from subcontracted plants to the company plants are included in the subcontracting costs.

The last variable cost concerns transportation. For this category of data, we use the formula proposed by Melkote and Daskin (2001). Each facility in the logistic network is represented as a node on a 100×100 grid where their x and y coordinate are randomly chosen with a uniform distribution in the interval $[0, 100]$. Then, the transportation cost between two points is proportional to the Euclidean distance between them.

References

- Ambrosino, D., Scutellà, M.G., 2005. Distribution network design: New problems and related models. *European Journal of Operational Research* 165, 610–624.
- Ballou, R.H., 1968. Dynamic warehouse location analysis. *Journal of Marketing Research* 5, 271–276.
- Beamon, B.M., 1998. Supply chain design and analysis: Models and methods. *International Journal of Production Economics* 55, 281–294.
- Canel, C., Khumawala, B.M., Law, J., Loh, A., 2001. An algorithm for the capacitated, multi-commodity multi-period facility location problem. *Computers & Operations Research* 28, 411–427.
- Cordeau, J.-F., Pasin, F., Solomon, M.M., 2006. An integrated model for logistics network design. *Annals of Operations Research* 144 (1), 59–82.
- Cortinhal, M.J., Captivo, M.E., 2003. Upper and lower bounds for the single source capacitated location problem. *European Journal of Operational Research* 151, 333–351.
- Daskin, M., Snyder, L., Berger, R., 2005. Facility location in supply chain design. In: Langevin, A., Riopel, D. (Eds.), *Logistics Systems: Design and Optimization*, pp. 39–65.
- Dias, J., Captivo, M.E., Climaco, J., 2006. Capacitated dynamic location problems with opening, closure and reopening of facilities. *IMA Journal of Management Mathematics* 17 (4), 317–348.
- Filho, V.J.M.F., Galvão, R.D., 1998. A tabu search heuristic for the concentrator location problem. *Location Science* 6, 189–209.
- Hamer-Lavoie, G., Cordeau, J.-F., 2006. Un modèle pour la conception d'un réseau de distribution avec localisation, affectation et stocks. *INFOR* 44, 99–115.
- Hinojosa, Y., Puerto, J., Fernández, F.R., 2000. A multiperiod two-echelon multicommodity capacitated plant location problem. *European Journal of Operational Research* 123, 271–291.
- Jardin, A., Dejax, P., Péton, O., 2006. A dynamic model for the strategic design of complex supply chains. *Research Report 06/4/AUTO*, Ecole des Mines de Nantes-IRCCyN.
- Martel, A., 2005. The design of production–distribution networks: A mathematical programming approach. In: Geunes, J., Pardalos, P.M. (Eds.), *Supply Chain Optimization*. Springer, Berlin, pp. 265–306.
- Melachrinoudis, E., Min, H., 2000. The dynamic relocation and phase-out of a hybrid, two-echelon plant/warehousing facility: A multiple objective approach. *European Journal of Operational Research* 123, 1–15.
- Melkote, S., Daskin, M.S., 2001. Theory and methodology—Capacitated facility location/network design problems. *European Journal of Operational Research* 129, 481–495.
- Melo, M.T., Nickel, S., Saldanha da Gama, F., 2006. Dynamic multi-commodity capacitated facility location: A mathematical modelling framework for strategic supply chain planning. *Computers & Operations Research* 33, 181–208.
- Owen, S.H., Daskin, M.S., 1998. Strategic facility location: A review. *European Journal of Operational Research* 111, 423–447.
- Paquet, M., Martel, A., Desaulniers, G., 2004. Including technology selection decisions in manufacturing network design models. *International Journal of Computer Integrated Manufacturing* 17 (2), 117–125.
- Pirkul, H., Jayaraman, V., 1998. A multi-commodity, multi-plant, capacitated facility location problem: Formulation and efficient heuristic solution. *Computers Operations Research* 25, 869–878.
- Saldanha da Gama, F., Captivo, M.E., 1998. A heuristic approach for the discrete dynamic location problem. *Location Science* 6, 211–223.
- Scott, A.J., 1971. Dynamic location-allocation systems: Some basic planning strategies. *Environment and Planning* 3, 73–82.
- Syam, S.S., 2002. A model and methodologies for the location problem with logistical components. *Computers & Operations Research* 29, 1173–1193.
- Vila, D., Martel, A., Beauregard, R., 2006. Designing logistics networks in divergent process industries: A methodology and its application to the lumber industry. *International Journal of Production Economics* 102 (2), 358–378.



ELSEVIER

Contents lists available at ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

A DC programming heuristic applied to the logistics network design problem

Phuong Nga Thanh^a, Nathalie Bostel^b, Olivier Péton^{a,*}^a *École des Mines de Nantes, IRCCyN, 4 rue Alfred Kastler, B.P. 20722, 44307 Nantes Cedex 3, France*^b *IUT de Saint-Nazaire, IRCCyN, 58 rue Michel Ange, B.P. 420, 44606 Saint-Nazaire Cedex, France*

ARTICLE INFO

Article history:

Received 1 December 2009

Accepted 16 September 2010

Available online 12 October 2010

Keywords:

DC programming

Heuristics

Mixed integer linear programming

Logistics network design

ABSTRACT

This paper proposes a new heuristic method for the logistics network design and planning problem based on linear relaxation and DC (difference of convex functions) programming. We consider a multi-period, multi-echelon, multi-commodity and multi-product problem defined as a large scale mixed integer linear programming (MILP) model. The method is experimented on data sets of various size. The numerical results validate the efficiency of the heuristic for instances with up to several dozens facilities, 18 products and 270 retailers.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

An efficient configuration of a logistics network must enable the production and delivery of goods to customers at the lowest cost while satisfying a required service level. This topic has been the subject of numerous optimisation models and methods in the fields of operations management and operations research. Optimisation of supply chain design and strategic planning problems are often modelled by mixed integer linear programmes (MILPs). In these models, the location issues are often represented by binary variable decisions while the product flows along the logistics network are represented by continuous decision variables.

Several reviews have been published in the last 10 years. Beamon (1998) distinguishes models with deterministic data from those with stochastic data. Owen and Daskin (1998) clearly separate the static and dynamic models and list various supply chain performance measures. Sahin and Süral (2007) present a wide range of applications. Finally, Daskin et al. (2005) and Melo et al. (2009) propose an extensive review of location problems in the context of supply chain design and planning.

The present paper proposes an enhanced version of an LP-rounding heuristic for the design and planning of complex supply chains. The idea of LP-rounding is to solve the linear relaxation of some MILPs and to round the fractional variables in order to recover integer feasible solutions. Despite its poor formal guarantee of performance, linear relaxation is known to yield good lower bounds for some assignment or location problems (Benders and

van Nunen, 1983). LP-rounding methods have been recently applied to the general assignment problem (French and Wilson, 2007) or lot-sizing problems (Hardin et al., 2007). In the field of facility location or logistics network planning, linear relaxation-based heuristics have been used for the single facility location and more complex models (Levi et al., 2004).

In a recent paper, Thanh et al. (2010) proposed an LP-rounding method combined with correction procedures for the strategic planning of complex logistics networks. The method relies on successive linear relaxations of the original MILP. At each iteration, some facility location variables are set at 0 or 1, either directly by the linear relaxation or by some rounding procedures. The method produces a sequence of MILPs of decreasing size. After a large enough number of iterations, the resulting MILP can be eventually handled by a solver. The heuristic is able to produce feasible solutions for every tested instance thanks to efficient correction procedures. The observed average reduction in computation time compared to the solver is around 80% while the loss in the objective function is 1.5%. However, for larger instances, the size of the residual MILP at the last step hinders the efficiency of the heuristic.

The proposed improvement consists of integrating a DC programming step into this method. At each iteration, the optimisation problem is modelled as a DC (difference of convex functions) problem and solved with the DC algorithm (DCA). This algorithm is used as a local search to improve the current upper bound. The numerical experiments show that introducing this DC formulation to solve the relaxed problem speeds up the heuristic. It also yields solutions that rarely require the use of the correction procedures.

The remainder of this paper is organised as follows. Section 2 summarises the underlying mathematical model, which is extensively detailed in Appendix A. Section 3 recalls the main principles of DC programming and the DC algorithm. In Section 4, we introduce

* Corresponding author.

E-mail addresses: phuong-nga.thanh@emn.fr (P.N. Thanh), Nathalie.Bostel@univ-nantes.fr (N. Bostel), olivier.peton@emn.fr (O. Péton).

the DC-based heuristic and show how to apply it to the logistics network design problem. Numerical results are presented in Section 5.

2. A brief description of the model

We consider a supply chain composed of four layers: potential first tier suppliers, manufacturing centres (plants), distribution centres and retailers or final customers. These layers are represented in Fig. 1.

The logistics network is dedicated to the production and distribution of several products manufactured in one or several steps. The model under consideration can be viewed as a facility location problem with multiple layers, multiple commodities, multiple periods and additional constraints. The complete description of the MILP has been proposed in Thanh et al. (2008) and can be found in Appendix A. In this section, we only outline its main components. The objective function to be minimised is the sum of all the logistics costs over the time horizon considered. This includes five fixed costs (supplier selection, cost of opening, closing or enlarging facilities, cost of operating facilities) and four variable costs (production or subcontracting the production, storage, transportation between facilities). All these costs are linear and, without loss of generality, they are considered as time-independent.

The decision variables include binary variables that model strategic decisions and continuous variables that model product flow along the network.

The binary decision variables are the following:

- $x_i^t = 1$ if facility i is active at period t . Depending on the context, one variable x_i^t can represent a supplier, a plant or a distribution

centre. A supplier is said to be active if it delivers at least one raw material. A plant or a distribution centre is active if it is open in the corresponding period. As shown in Thanh et al. (2008), variables x_i^t , associated with plants and distribution centres, are the core of the problem and have a direct influence on a large part of the cost function.

- $y_{i,o}^t = 1$ if facility i is enlarged at period t by adding some optional capacity o . The optional capacities are modular extensions that may be appended to existing production or distribution centres in order to increase their capacity. For each facility, there is a list of available sizes of optional capacities.
- Variables $z_{s,p}^t$ and v_s^t concern supplier selection and supplier discounted costs. Since the heuristic focuses on location variables, we do not detail them.

The continuous decision variables are the following:

- $q_1^t(p,i,j)$ is the quantity of product p sent from facility i to facility j at period t ,
- $q_2^t(p,i)$ is the quantity of product p produced in plant i at period t ,
- $q_3^t(p,i)$ is the quantity of product p subcontracted by plant i at period t ,
- $q_4^t(p,j)$ is the quantity of product p held in distribution centre j at period t .

The set of constraints models realistic rules for the management of the whole network. It can be divided into four categories: demand satisfaction constraints, capacity limitation constraints, coherence constraints (for example only open facilities can produce) and integrity and non-negativity constraints.

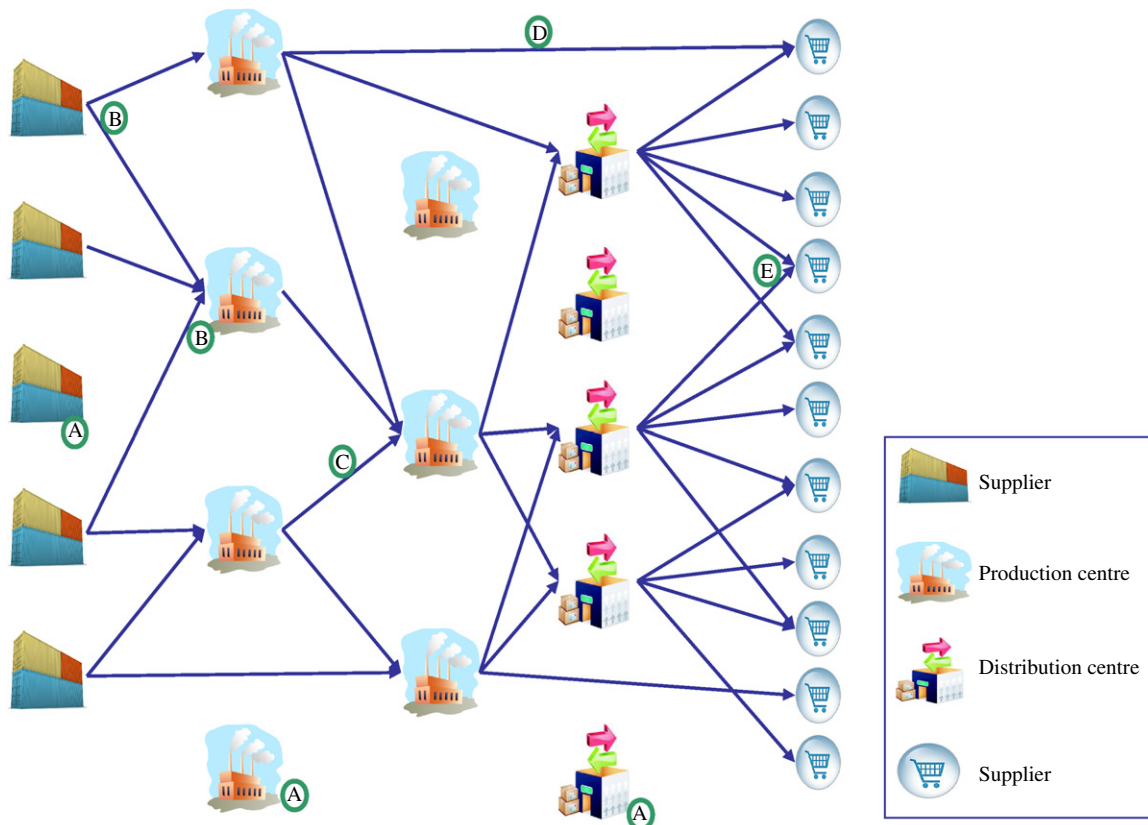


Fig. 1. The logistic network considered. (A) Some facilities may not be used. (B) Suppliers can deliver several manufacturing centres and manufacturing centres can be delivered by several suppliers. (C) Production of goods can be split among various manufacturing centres. In this case there must exist a link between the corresponding facilities. (D) There can exist direct deliveries from manufacturing centres to some important customers. (E) The customers can be delivered by various distribution centres (no single sourcing constraint).

3. DC programming and the DCA

3.1. The simplified DCA scheme

DC programming consists of reformulating an optimisation problem as the difference of two convex functions (DC = difference of convex functions) and solving it with a DC algorithm (DCA). A DC problem is defined as

$$\text{Min}\{g(x)-h(x)|x \in \mathbb{R}^n\} \quad (P_{DC})$$

where g and h are lower semi-continuous proper convex functions on \mathbb{R}_n . This definition is very wide and most real optimisation problems can be modelled with a DC approach even if they are non-convex. The DC algorithm (DCA) is an efficient method to solve DC problems. It was introduced by Pham in its preliminary form (Pham and El Bernoussi, 1986) and has been extensively developed by Pham and Le Thi since 1994. In the remainder of this section we describe the main principles of the DCA and explain how to apply it to MILPs. The interested reader will find more details in Le Thi and Pham (2005, 2008).

The DCA is a local method based on subgradients and on duality optimisation in DC programming. The duality theory was introduced by Toland (1978) and defined the dual problem of (P_{DC}) as

$$\text{Min}\{h^*(y)-g^*(y)|y \in \mathbb{R}^n\} \quad (D_{DC})$$

where $g^*(y) = \sup\{\langle x,y \rangle - g(x)|x \in \mathbb{R}^n\}$ is the conjugate function of g . We observe the perfect symmetry between primal and dual DC problems: the dual of (D_{DC}) is exactly (P_{DC}) .

The DCA is based on an iterative construction of two sequences $\{x^k\}$ and $\{y^k\}$, candidates for optimal solutions of primal and dual problems, respectively. These two sequences $\{x^k\}$ and $\{y^k\}$ are determined in the way that $\{x^{k+1}\}$ (resp. $\{y^{k+1}\}$) is a solution to the convex problem (P_k) (resp. (D_k)) defined by

$$\text{inf}\{g(x)-h(x^k)-\langle x-x^k,y^k \rangle |x \in \mathbb{R}^n\} \quad (P_k)$$

$$\text{inf}\{h^*(y)-g^*(y^{k-1})-\langle y-y^{k-1},x^k \rangle |y \in \mathbb{R}^n\} \quad (D_k)$$

At each iteration, one replaces in the primal DC problem (P_{DC}) the second component h by its affine minorisation $h_k(x) := h(x^k) + \langle x-x^k,y^k \rangle$ in the neighbourhood of x^k to produce the convex problem (P_k) whose solution set is nothing but $\partial g^*(y^k)$. Likewise, the second DC component g^* of the dual DC problem (D_{DC}) is replaced by its affine minorisation $(g^*)_k(y) := g^*(y^k) + \langle y-y^k,x^k \rangle$ in the neighbourhood of y^k to obtain the convex problem (D_k) whose $\partial h(x^{k+1})$ is the solution set. The DCA performs a double linearisation with the help of the subgradients of h and g^* and the DCA then yields the following scheme:

$$y^k \in \partial h(x^k), \quad x^{k+1} \in \partial g^*(y^k)$$

The simplified DCA scheme is then described by Algorithm 1.

Algorithm 1. The simplified DCA scheme

Initialisation: set $x^0 \in \mathbb{R}^n$.

Iteration k:

Determine $y^k \in \partial h(x^k)$.

Find $x^{k+1} \in \partial g^*(y^k)$.

if some stopping criterion is satisfied **then**

STOP

else

$k \leftarrow k+1$.

end if

The DCA has been successfully applied to many large-scale—smooth or non-smooth—non-convex optimisation problems in various fields of applied sciences, including strategic supply chain

design (Le Thi et al., 2007) and continuous location (Drezner and Nickel, 2009).

DC functions generally have many potential DC decompositions in functions g and h . The choice of this decomposition has a crucial impact on the quality of the DCA. For a given DC problem, the choice of optimal DC decompositions is still open. Of course, this depends strongly on the very specific structure of the problem being considered. One tries in practice to choose g and h such that sequences $\{x^k\}$ and $\{y^k\}$ can be easily calculated, i.e. either they are in explicit form or their computations are inexpensive. It is worth noting that the DCA depends greatly on components g and h but not the DC function $g-h$ itself.

3.2. The DCA for mixed integer problems

In this paper, the DCA is used to solve a mixed integer problem. Let us consider an MILP of the following general form:

$$\begin{aligned} \text{Min} \quad & f(x,y) = c^T x + d^T y \quad (\text{MILP}) \\ & Ax + By \leq E \\ & x \in \{0,1\}^n \\ & y \in \mathbb{R}^p \end{aligned}$$

Solving this problem with the DCA requires some reformulation. The first operation consists of transforming it into a DC problem. This is the goal of the exact penalty theorem (Le Thi et al., 1999).

Theorem 1 (Exact penalty theory). Let C denote a non-empty bounded convex polyhedron; f a concave function defined on C and p a non-negative function concave on C . There exists $\tau_0 \geq 0$ such that for all $\tau > \tau_0$, problems (P) and (P_τ) have the same set of feasible solutions and the same optimal value:

$$\begin{aligned} \text{Min}\{f(x,y)|(x,y) \in C, p(x) \geq 0\} \quad (P) \\ \text{Min}\{f(x,y) + \tau p(x)|(x,y) \in C\} \quad (P_\tau) \end{aligned}$$

p is called the penalty function and τ is called the coefficient of the penalty function.

In order to reformulate (MILP), the integrality constraints on x are relaxed and integrated to the objective function by way of the penalty function.

$$\begin{aligned} \text{Min} \quad & f(x,y) = c^T x + d^T y + \tau p(x) \\ & Ax + By \leq E \\ & 0 \leq x_i \leq 1 \quad \forall i = 1, \dots, n \\ & y \in \mathbb{R}^p \end{aligned}$$

The problem is thus reformulated as the minimisation of the non-linear function $f(x,y) = c^T x + d^T y + \tau p(x)$ over the convex set $K = \{(x,y)|Ax + By \leq E, 0 \leq x_i \leq 1, y \in \mathbb{R}^p\}$. The usual penalty functions associated with binary variables x_i are $p(x) = \sum_i \text{Min}\{x_i, 1-x_i\}$ or $p(x) = \sum_i x_i(1-x_i)$.

$f(x,y)$ can now be expressed as a difference of two convex functions g and h following the usual decomposition (Le Thi and Pham, 2005, 2008):

$$\begin{aligned} g(x,y) &= \chi_k(x,y) \\ h(x,y) &= -c^T x - d^T y - \tau p(x) \end{aligned}$$

where

$$\chi_k(x,y) = \begin{cases} 0 & \text{if } (x,y) \in K \\ +\infty & \text{otherwise} \end{cases}$$

4. Description of the heuristic

We first give the general outline of the heuristic. The important procedures are then described in the relevant subsections.

4.1. Outline of the heuristic

The main contribution of the heuristic is to combine LP-rounding and DC programming. DC programming is used to build the sequence of relaxed problems that are solved with the DCA. At each iteration, a rounding procedure is used to freeze as many decision variables as possible. In practice variable that show a strong tendency toward 0 or 1 are frozen. Once frozen, the binary variables are considered as constants at subsequent iterations. When the residual number of binary variables is small enough, the residual MILP is solved by an exact method.

The heuristic is depicted in Fig. 2.

The numbers in the circles refer to the number of the subsection where the corresponding operations are detailed.

4.2. Correction procedure

The correction procedure is applied to decision variables x_i^t and x_j^t . Its objective is to restore the feasibility of the current solution by changing the status of some facilities (production or distribution centres). It is based on the feasibility pump heuristic for mixed integer problems (Fischetti et al., 2005; Bertacco et al., 2007). Given an integer unfeasible solution, this heuristic seeks the nearest feasible solution whether it is integer or not. We minimise the distance $\delta_{\tilde{x}}(x) = \sum_{i \in I} |x_i - \tilde{x}_i|$ between two vectors \tilde{x} (the given integer solution) and x (any feasible solution). Function $\delta_{\tilde{x}}(x)$ is not linear. However for special cases, when $\tilde{x}_i \in \{0; 1\}$, it can be linearised by $\delta_{\tilde{x}}(x) = \sum_i \tilde{x}_i(1 - 2x_i) + x_i$.

We have

$$\delta_{\tilde{x}_i}(x_i) = \begin{cases} x_i & \text{if } \tilde{x}_i = 0 \\ 1 - x_i & \text{if } \tilde{x}_i = 1 \end{cases}$$

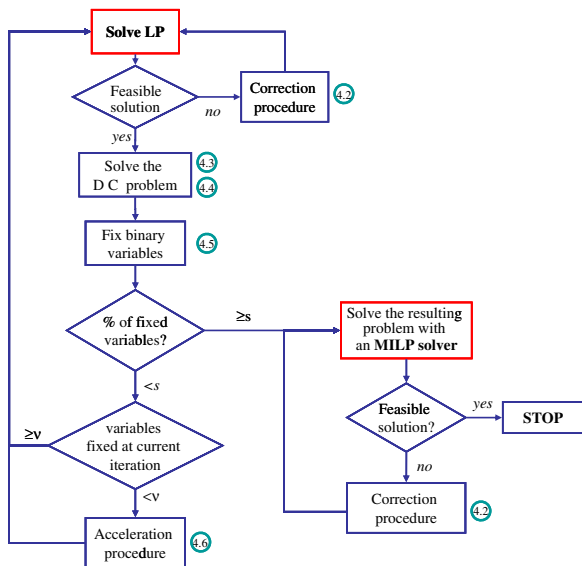


Fig. 2. Main scheme of the heuristic. The objective of Step 1 is to round variables x_i^t to 0 or 1 until enough variables have been frozen. The following operations are repeated until the required percentage of frozen variables is reached. If the initial point is unfeasible, the correction procedure is used. Otherwise, it can be used directly in the DCA. Then the MILP is reformulated as a DC problem and the DC algorithm is applied. All binary variables x_i whose relaxed values are found to be exactly 0 or 1 are fixed. Then the variables that show a strong tendency toward 0 or 1 are rounded. Whenever needed, the acceleration procedure is called in order to reach a target level s . In Step 2, the residual MILP is solved by an MILP solver. If the solution is unfeasible, the correction procedure is called.

Since every variable is not rounded at the current iteration, the feasibility pump has to consider a subset of active variables. In practice, we solve

$$\text{Min} \delta_{\tilde{x}}(x) = \sum_{i \in I} (\tilde{x}_i(1 - 2x_i) + x_i) \text{Ind}(i)$$

where $\text{Ind}(i) = 1$ if the variable \tilde{x}_i is fixed, and 0 otherwise. For the obtained feasible solution x , we fix all binary variables whose values are exactly 0 or 1.

Several authors have pointed out some potential problems with the feasibility pump. First, the procedure can be caught in a cycle. This means that the same sequence of integer and LP-feasible points is visited over and over again. Second, the progress in driving the integer points toward feasibility might be very slow (Achterberg and Berthold, 2007). In the present case, unfeasibility is generally caused by bad rounding of a very small number of decision variables. The instances we solve have a large number of feasible solutions so that a few iterations of the original feasibility pump are always enough to retrieve feasibility.

4.3. Formulating the DC problem

Let us denote variables x_s^t for the selection of customers, variables x_i^t for the location of production centres and variables x_j^t for the location of distribution centres. Applying the theory of the exact penalty, we relax the integrality constraints and add a penalty function p in the objective function. The penalty function is defined as

$$p(x, y, z, v) = \tau \sum_{s \in S} \min\{x_s^t, 1 - x_s^t\} + \tau_i \sum_{i \in I} \min\{x_i^t, 1 - x_i^t\} + \tau_j \sum_{j \in J} \min\{x_j^t, 1 - x_j^t\} + \tau \sum_{i \in I \cup J, o \in O} \min\{y_{i,o}^t, 1 - y_{i,o}^t\} + \tau \sum_{s \in S, p \in P_r} \min\{z_{s,p}^t, 1 - z_{s,p}^t\} + \tau \sum_{s \in S} \min\{v_s^t, 1 - v_s^t\}$$

where $\tau_i \geq 0$ and $\tau_j \geq 0$ are the coefficient associated with the location binary variables for the production centres and the distribution centres, respectively. The coefficient $\tau \geq 0$ is associated with other binary variables. We define dedicated penalty coefficients (τ_i and τ_j) for the variables x_i^t and x_j^t since they have a major role in the solution definition and deserve special attention. These coefficients are increased during the acceleration procedure; this acts as a diversification procedure when defining the initial solution for the DCA.

Let (C) denote the feasible set of constraints in the original MILP. The modified mathematical model with a DC form is the following:

$$\begin{aligned} \text{Min } & z + p \\ \text{s.t. } & (C) \\ & 0 \leq x_i^t, y_{i,o}^t, z_{p,s}^t, v_s^t \leq 1 \\ & q_1, q_2, q_3, q_4 \geq 0 \end{aligned}$$

Since variables $x_i^t, y_{i,o}^t, z_{p,s}^t$ and v_s^t have been turned to continuous variables, the set of constraints of this DC formulation defines some convex set K .

We apply the usual decomposition of the objective function for MILPs into two convex functions (Le Thi and Pham, 2005, 2008) g and h . To simplify the presentation of the two sequences of the DCA at iteration k , let us denote $X^k = (x, y, z, v, q_1, q_2, q_3, q_4)^T$. We define

$$g(X^k) = \chi_K(X^k)$$

$$h(X^k) = -z - p(x, y, z, v)$$

where χ_K is the indicator function of set K .

$$\chi_K(X^k) = \begin{cases} 0 & \text{if } (X^k) \in K \\ +\infty & \text{otherwise} \end{cases}$$

4.4. Calculation of the DCA sequence (x^k and y^k)

Since the DCA is a local search method, the initial point x^0 plays a key role in the quality of the solution. A suggestion for finding a good candidate is generally to minimise the penalty function with the DCA: $\text{Min}\{p(x)|x, y \in K\}$. We use the solution given by the linear relaxation of the current MILP. If this solution is feasible, it can be used directly as a starting point. Otherwise, we resort to a correction procedure that retrieves a feasible solution.

4.4.1. Calculation of the Y^k sequence

The Y^k sequence is computed from the X^k as the subgradients of $h(X^k)$. We have $Y^k = \partial h(X^k) = -\partial(z) - \partial(p)$.

The calculation of $\partial h(X^k)$ is divided into two parts: the calculation of the subgradient of the penalty function p and those of the initial objective function z . We detail these calculations of Y^k in Appendix B.

4.4.2. Calculation of the X^{k+1} sequence

Proposition 1. X^{k+1} is the solution of the linear problem $\text{min}\{-\langle Y^k, X \rangle | X \in C\}$.

Proof. From equation (P_k) , X^{k+1} is the solution of the convex problem

$$\begin{aligned} &\text{Min}\{g(X) - h(X^k) - \langle X - X^k, Y^k \rangle | X \in C\} \\ &= \text{Min}\{g(X) - \langle X, Y^k \rangle - h(X^k) + \langle X^k, Y^k \rangle | X \in C\} \quad \square \end{aligned}$$

For a given X^k and Y^k , the expression $-h(X^k) + (X^k)^T Y^k$ is constant. The optimal solution of (P_X^{k+1}) is then determined by

$$\text{Min}\{g(X) - \langle X, Y^k \rangle\}$$

In our decomposition, we have $g(X) = \chi_C(X)$. The problem (P_X^{k+1}) is thus equivalent to the linear problem

$$X^{k+1} \in \text{Argmin}\{-\langle Y^k, X \rangle | X \in C\}$$

4.5. Binary variable fixation procedure

After solving the linear relaxation, the heuristic fixes all binary variables whose relaxed values are exactly 0 or 1 or show a strong

tendency toward 0 or 1. In practice, we define asymmetric threshold values ε_0 and ε_1 so that

- if some variable x_i is such that $0 \leq x_i \leq \varepsilon_0$, x_i is set at 0.
- if some variable x_i is such that $\varepsilon_1 \leq x_i \leq 1$, x_i is set at 1.

Once fixed, the binary variables are considered as constants at subsequent iterations.

4.6. Acceleration procedure

One critical aspect of the heuristic is the need to diversify the starting point of the DCA. When few decision variables are fixed at a given iteration, there is the risk of initialising the DCA very close to the local optimum of the previous iteration. In this case, the heuristic may stagnate without reaching the threshold s for the level of fixed variables. The acceleration procedure detailed in Algorithm 2 aims to diversify the choice of initial points, and increase the probability of improving the local optimum found by the DCA.

Algorithm 2. Acceleration procedure

Arguments:

- τ_i : parameter for the penalty associated with the variables x_i
- α : multiplier of τ_i ($\alpha > 1$)
- γ : number of variables that are fixed at the current iteration
- v : minimum number of fixed variables to be exempted from the acceleration procedure
- η : threshold to fix a variable at 1

if $\gamma < v$ **then**

- x_i = The largest fractional variable
- \bar{x}_i = The variables related to variable x_i

if $x_i > \eta$ **then**

- $x_i \leftarrow 1$
- $\bar{x}_i \leftarrow 1$

else

- $\tau_i \leftarrow \alpha \cdot \tau_i$ ($\alpha > 1$)

end if

end if

The algorithm forces the opening of supplementary factories and/or distribution centres. Priority is given to the facilities associated with the largest fractional value x_i . Nevertheless, we

Table 1
Description of the test instances.

Family	Suppliers	Plants	Distribution centres	Customers	Products	Binary variables	Location variables
S1	15	10	5	100	10	750	75
S2	15	10	5	130	10	750	75
S3	17	12	5	130	12	1020	85
S4	17	12	7	160	12	1060	95
S5	19	14	7	160	14	1180	105
M1	19	14	7	180	14	1275	105
M2	19	14	7	180	16	1275	105
M3	21	16	9	180	16	1445	125
M4	21	16	9	210	16	1445	125
M5	23	18	9	210	16	1575	135
L1	23	18	9	240	16	1575	135
L2	23	18	11	240	18	1845	145
L3	25	20	11	240	18	1995	155
L4	25	20	13	270	18	2035	165
L5	27	22	13	270	18	2185	175

Column 1 indicates the family of the instance. **S, M** and **L** stand for small, medium and large size, respectively. Each family includes 30 instances. Columns 2–6 describe the main characteristics of every family: number of suppliers, plants, distribution centres, customers and products. Column 7 shows the number of binary variables in the corresponding MILP. Column 8 details the number of binary variables representing the location decisions.

wish to fix at 1 only the variables whose value exceeds a given threshold η . The idea is to avoid degrading the objective function too much. If no variable can be fixed at 1, the value of the coefficient τ_i is increased. These coefficients give more weight to the penalty function and complicating variables must be pushed more strongly toward bounds at the subsequent iterations. Note that setting a variable x_i^t at 1 impacts on some *related variables*. For example, the model imposes that, after opening, one plant cannot be closed before the end of the planning horizon. Thus, if variable x_i^t is set at 1, every variable $x_i^{t'}$, with $t' > t$ is also set at 1.

Algorithm 2 only applies to variables that have the largest influence on the whole solution, namely variables x_i^t (for plant location) and x_j^t (for distribution centre location).

5. Computational results

The heuristic method is implemented in C language while the linear relaxations and the residual MILP in Step 2 are solved by the solver Xpress-MP (2007). In addition, we solve every instance independently with Xpress-MP (with optimised parameters) in order to compare the performance with the heuristic method. The experiments are performed on a Pentium IV, 3.2 GHz processor with 1 GHz of RAM.

5.1. Benchmark instances

We generated a bench of 450 test instances regrouped into 15 families. Families S1–S5 (small-size instances) consider networks with 15–21 facilities to locate and 100–160 customers to deliver to; families M1–M5 (medium-size instances) handle 21–27 facilities and about 200 customers; families L1–L5 (large-size instances) reach up to 35 facilities and 300 customers. All instances have an identical strategic time horizon of five periods (of typically one year). Every family includes 30 instances whose characteristics are detailed in Table 1.

The instances are generated following the ideas of various authors. The customers' demands derive from Melo et al. (2006); the generation of the graph representing the logistics network follows a procedure taken from Cordeau et al. (2006); the capacity

Table 2
Results of the DC based heuristic for small-size (S) instances.

Family	Avg. time (in seconds)	Step 1 (%)	Corrections step 1		Corrections step 2	
			Instances	Iter.	Instances	Iter.
S1 ₁	32	69	2	1	0	0
S1 ₂	34	71	2	1	0	0
S1 ₃	35	75	3	1.3	0	0
S2 ₁	33	68	1	1	0	0
S2 ₂	58	67	2	1.5	0	0
S2 ₃	60	70	1	2	0	0
S3 ₁	72	67	1	1	0	0
S3 ₂	92	70	3	1	0	0
S3 ₃	102	69	2	2	0	0
S4 ₁	147	55	0	0	0	0
S4 ₂	211	46	1	1	0	0
S4 ₃	219	48	1	2	0	0
S5 ₁	262	69	2	1.5	0	0
S5 ₂	266	65	1	1	0	0
S5 ₃	272	65	2	1	0	0

The table presents the results for families S1–S5 and every type of instance (loose, medium or tight capacity). Column 2 is the average computation time of the heuristic (in seconds). Column 3 shows the percentage of computation time spent on Step 1. Columns 4 and 5 report the number of instances (out of 10) for which the correction procedure is called at Step 1, and the average number of iterations of this procedure, respectively. Columns 6 and 7 report the same information for Step 2.

of the facilities comes from Melkote and Daskin (2001); the principle of transition phases after opening or before closing a facility and the variable costs are inspired from Melachrinoudis and Min (2000); finally the fixed costs follow the same format as in Cortinhal and Captivo (2003). Every instance has been checked as being feasible.

In order to balance easy and difficult instances, the families are decomposed into three sub-families with a decreasing load/capacity ratio: 10 instances with a tight capacity (Type 1), 10 instances with an average load/capacity ratio (Type 2) and 10 instances with a loose capacity (Type 3). For example, for family S 1, these sub-families are denoted S 1₁, S 1₂ and S 1₃, respectively.

The complete details about the generation of instances can be found in Thanh (2008).

5.2. Parameter setting

The heuristic is based on eight parameters that are used for all the numerical results reported.

Step 1 of the heuristic is controlled by two parameters ϵ_0 and ϵ_1 . The threshold value ϵ_0 has been set at 0.03 and ϵ_1 at 1. The level s of desired binary variables determines the number $5 \times (1-s) \times \mathcal{I} \times \mathcal{J}$ of decision

Table 3
Results of the DC based heuristic for medium-size (M) instances.

Family	Avg. time (in seconds)	Step 1 (%)	Corrections step 1		Corrections step 2	
			Instances	Iter.	Instances	Iter.
M1 ₁	363	62	2	3	0	0
M1 ₂	383	62	0	0	0	0
M1 ₃	438	57	2	1	0	0
M2 ₁	443	75	2	1.5	0	0
M2 ₂	450	49	1	1	0	0
M2 ₃	456	62	2	1.5	0	0
M3 ₁	485	56	0	0	0	0
M3 ₂	501	54	0	0	0	0
M3 ₃	542	51	1	1	1	1
M4 ₁	484	67	1	2	0	0
M4 ₂	514	67	2	2.5	0	0
M4 ₃	595	54	1	2	0	0
M5 ₁	572	63	2	1.5	0	0
M5 ₂	588	56	1	1	0	0
M5 ₃	641	54	1	1.5	0	0

Table 4
Results of the DC based heuristic for large-size (L) instances.

Family	Avg. time (in seconds)	Step 1 (%)	Corrections step 1		Corrections step 2	
			Instances	Iter.	Instances	Iter.
L1 ₁	634	54	1	1	0	0
L1 ₂	685	51	2	2	1	1
L1 ₃	775	52	0	0	0	0
L2 ₁	763	42	0	0	0	0
L2 ₂	1101	43	1	1	1	1
L2 ₃	1241	35	0	0	0	0
L3 ₁	1199	43	0	0	0	0
L3 ₂	1218	38	0	0	0	0
L3 ₃	1313	33	1	1	0	0
L4 ₁	1204	34	0	0	0	0
L4 ₂	1269	38	0	0	0	0
L4 ₃	1338	31	1	1	0	0
L5 ₁	1285	38	1	1	0	0
L5 ₂	1295	39	2	1	0	0
L5 ₃	1569	38	1	1	0	0

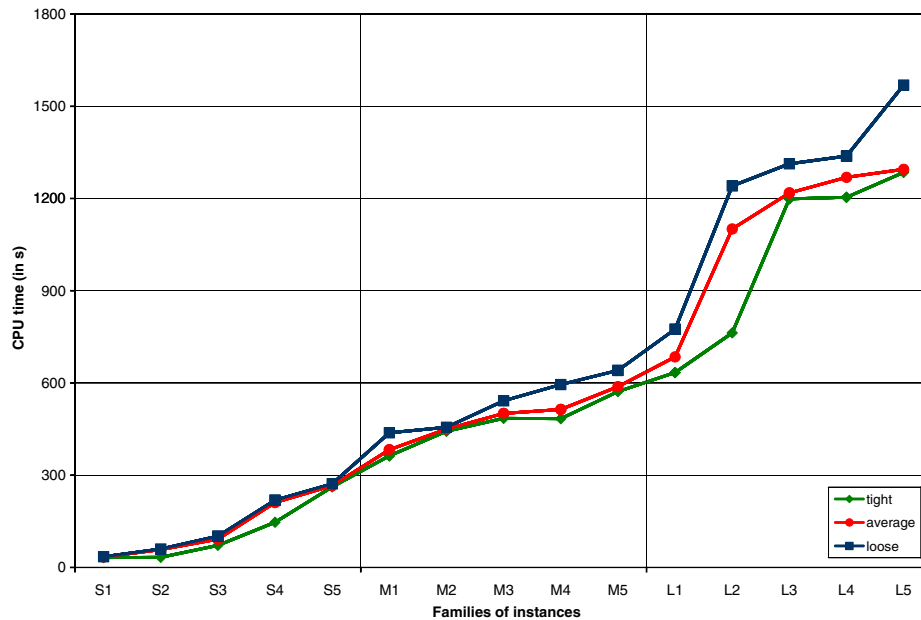


Fig. 3. Computation time of the DC based heuristic. The horizontal axis represents each family of instances. The vertical axis represents the computation times (in seconds). The points represent the average results over the 10 instances of each type and each family of instances.

Table 5 Efficiency of the heuristic on S instances.

Family	t_s	t_H	Δ_t (%)	Δ_z (%)	ρ_z (%)
S1 ₁	65	32	-29	0.5	0.48
S1 ₂	89	34	-49	0.6	0.05
S1 ₃	105	35	-44	2.2	2.03
S2 ₁	74	33	-14	0.3	0.01
S2 ₂	158	58	-53	1.3	1.77
S2 ₃	349	60	-12	2	3.3
S3 ₁	150	72	-49	0.2	0.23
S3 ₂	219	92	-39	0.3	-0.55
S3 ₃	386	102	-25	1	0.44
S4 ₁	473	147	-76	0.16	-3.94
S4 ₂	643	211	-70	0.83	0.77
S4 ₃	796	219	-63	1.32	1.04
S5 ₁	837	262	-50	0.28	-0.26
S5 ₂	1157	266	-62	0.17	-0.62
S5 ₃	1419	272	-68	0.69	0.16

Columns 2 and 3 show the computation time t_s of the MILP solver (in seconds) and the computation time t_H of the heuristic method (in seconds), respectively. The relative difference $\Delta_t = (t_H - t_s) / t_s \times 100$ is reported in Column 4. As the value of the objective function has no intrinsic significance, we report the relative difference Δ_z between both objectives in Column 5. It is calculated as $\Delta_z = (z_H - z_S) / z_S \times 100$, where z_H is the value of the objective function found by the heuristic and z_S is the value of the objective function found by the MILP solver. The last column compares the result of both methods with the same allocated computation time. We calculate the value $\rho_z = (z_H - z'_S) / z'_S \times 100$, where z'_S is the value found by the solver after the same computation time as the heuristic.

variables in the MILP solved at Step 2, where \mathcal{I} is the number of production centres, \mathcal{D} the number of distribution centres and 5 the number of periods considered in the model. Thus s must increase when the problem size increases. We set s at 0.6 for families S1–S5, at 0.7 for families M1–M5 and at 0.8 for families L1–L5.

The real parameters τ , τ_i and τ_j play an important role in the value of the penalty function p . We fix them in such a way as to balance the main objective function z and the penalty p . Thus, large values are needed for these parameters. We set $\tau_i = \tau_j = \tau = 1000$. Parameters τ_i and τ_j are then modified through multiplier α during the acceleration procedure (Algorithm 2). Parameter α is set at 1.2 for families S1–S5, at 2 for families M1–M5 and at 10 for families L1–L5.

Table 6 Efficiency of the heuristic on M instances.

Family	t_s	t_H	Δ_t (%)	Δ_z (%)	ρ_z (%)
M1 ₁	1335	363	-62	0.06	-2.38
M1 ₂	1617	383	-62	0.31	-0.47
M1 ₃	2133	438	-75	0.98	-2.29
M2 ₁	3481	443	-82	0.18	-2.17
M2 ₂	3872	450	-80	0.45	-3.21
M2 ₃	4682	456	-87	1.8	-3.6
M3 ₁	4349	485	-75	0.16	-1.69
M3 ₂	4533	501	-83	-0.005	-5.55
M3 ₃	4641	542	-83	0.45	-3.38
M4 ₁	6289	484	-90	1.43	-0.17
M4 ₂	6379	514	-85	2	-1.89
M4 ₃	6442	595	-81	1.03	-2.83
M5 ₁	6967	572	-85	1.15	-0.13
M5 ₂	2h	588	-86	2	-0.5
M5 ₃	2h	641	-88	0.34	-7.7

No instance of families M5₂ and M5₃ could be solved to optimality by the solver. The calculation was thus stopped after 2 hours (written 2 h).

The acceleration procedure uses two more parameters. η is a threshold value to fix a variable at 1. It is set at 0.1. ν is the minimum number of variables that must be fixed during an iteration. It is set at 10.

The parameter setting was carried out through designed experiments. The selected parameters ensure a good compromise between the quality of the solution and the computation time.

5.3. Numerical results

We report the results of the heuristic in Tables 2–4. Columns 4–7 refer to the correction procedure. Since this procedure may be time consuming, it is better not to run many iterations.

We observe that the number of calls to the correction procedure is quite limited. The maximal number of calls for the whole heuristic (Step 1 + Step 2) is only 3. Moreover, a few iterations of this procedure are enough to restore feasibility. Another remark is that the percentage of time spent on Step 1 decreases when the difficulty of the instances increases. This can be explained by the growth of the residual MILP at Step 2. Only 1 out of the 450 MILPs at Step 2 could not be solved to

optimality by the solver. If larger datasets were considered, this number would certainly increase. Increasing the value of parameter s for large datasets would be necessary with the risk of taking wrong decisions in the acceleration procedure at the end of Step 1 and thus degrading the objective function.

Fig. 3 compares the computation time for type of instance (loose, medium or tight capacity). It tends to increase quite linearly when the instance size grows and is spread out between 100 and 1800s. This is a significant improvement compared to the heuristic proposed in Thanh et al. (2010). The instances with loose capacity appear to be slightly more difficult to solve.

5.4. Evaluation of the heuristic's performance

To evaluate the quality of the heuristic, we compare it with the results obtained by the solver Xpress-MP. For some medium-size and almost all large instances, the solver is unable to provide an

optimal solution within a reasonable time of computation. We allocate a limited computation time to the solver: 1 h for the S1–S5 instances, 2 h for the M1–M5 instances and 3 h for the L1–L5 instances. We propose these limitations after observing the behaviour of the objective function in our experiments. In the first minutes of computation, the objective function quickly decreases and then stabilises. For the largest instances, improvements are seldom observed after 2–3 h of computation. Moreover, these improvements are highly unpredictable. Thus, we decide to stop the computations after a predefined time limit. If no optimal solution is found within this limit, the best upper bound obtained is used for the calculation of the deviation.

When we ignore the optimal solution for an instance, we measure the relative gap between the result of the heuristic and the best feasible solution found by the solver. We cannot use the lower bounds since they are not always provided by the solver.

Tables 5–7 compare the computation time and the value of the objective function obtained with the solver and the heuristic.

The results show that the DCA-based heuristic is faster than the solver for every instance. This gap increases with the size of the instances. For families S1–S3, Δ_t is about –30% on average, –60% for families S4 and S5, and –90% for families L1–L5. However, this speed-up is accompanied by a degradation of the objective function. In the worst cases, the optimality gap is 3% at most. For 33 out of the 45 families, the gap is less than 1%. For some families (M3₂, L5₂, L5₃), the heuristic dominates the solver approach in the sense that the computation time and the value of the objective are improved simultaneously. If we stop the solver after the same computation time as the heuristic and compare the solution, there is not much difference as far as the small or medium size instances are concerned. However, for large-size instances the heuristic outperforms the solver for all instances, especially the largest ones (family L5₃). We also notice that the heuristic performs particularly well on the most difficult instances (with loose capacity constraints).

Fig. 4 gives a graphical interpretation of Table 7. The figure clearly shows that the heuristic generally improves the computation time from 60% to 100%. The price to pay is a slight degradation of the objective function (between 0% and 2%)

Table 7 Efficiency of the heuristic on L instances.

Family	t_s	t_H	Δ_t (%)	Δ_z (%)	ρ_z (%)
L1 ₁	3h	634	–96	3	–0.56
L1 ₂	3h	685	–88	1.2	–9.46
L1 ₃	3h	775	–92	0.93	–36.4
L2 ₁	3h	763	–89	0.72	–3.07
L2 ₂	3h	1101	–80	0.92	–3.83
L2 ₃	3h	1241	–89	0.48	–7.15
L3 ₁	3h	1199	–84	0.94	–4.7
L3 ₂	3h	1218	–83	1.03	–6.1
L3 ₃	3h	1313	–87	0.31	–5.4
L4 ₁	3h	1204	–87	0.9	–5.98
L4 ₂	3h	1269	–75	0.2	–10.53
L4 ₃	3h	1338	–85	0.87	–21.1
L5 ₁	3h	1285	–86	0.95	–3.89
L5 ₂	3h	1295	–85	–0.03	–39.84
L5 ₃	3h	1569	–79	–3	–57.2

Only one instance in family L2₁ could be solved to optimality. The average CPU time for this family is still very close to 3 hours (written 3 h for simplicity).

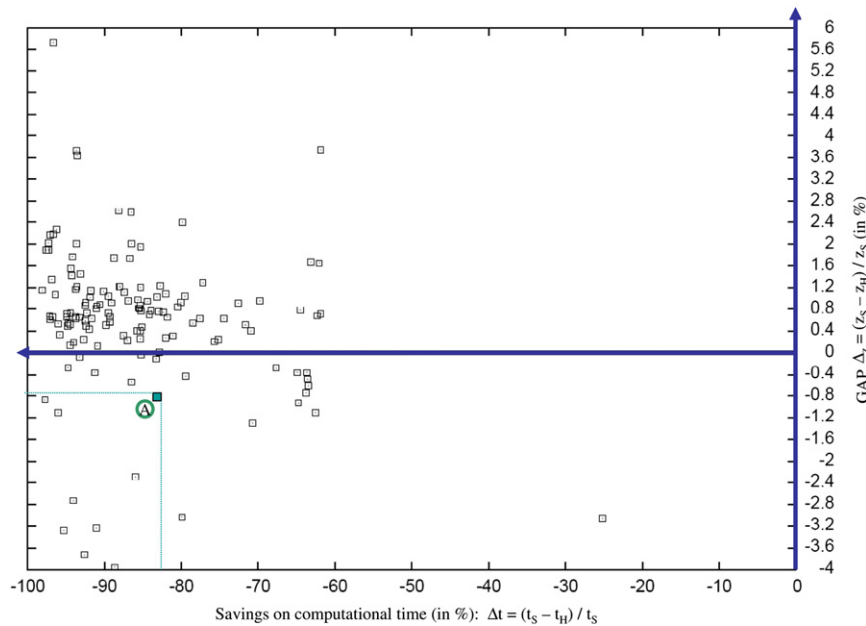


Fig. 4. Performance of the heuristic on L instances. The horizontal axis represents the relative gap in computational time Δ_t between the heuristics and the solver. For example –50% means that the heuristics is twice faster. The vertical axis represents the relative gap Δ_z on the objective function. Each small square represents one of the 150 instances of families L1–L5. Let us consider the example of instance A: the savings on computational time is 82.5% while the objective function obtained with the heuristic is 1.8% better than the one obtained with the solver after 3 h of computation.

Although efficient on average, the heuristic still has some problems in solving a few difficult instances. We observe cases with a limited improvement in the computation time or a relatively large optimality gap. The robustness of the heuristic is still an issue. We believe that combining the DC approach with a more efficient relaxation technique (for example Lagrangean relaxation) would overcome this problem.

6. Conclusion and further research

We have proposed a new heuristic method for the logistics network design and planning problem. This method improves the LP-rounding heuristic method that was proposed in Thanh et al. (2010). For the largest instances, the LP-rounding procedure reached its practical limits and the improvement compared to a commercial solver was not significant enough. Therefore, we have designed a new heuristic to improve the computation performance, especially on large-scale instances. The proposed method is also based on LP-rounding where the problem is reformulated in DC programming and solved with the DCA at each iteration. The method focuses on location variables, which are rounded by using tailored rules. An acceleration procedure is also proposed to enhance the efficiency of the integer variables rounding.

We have tested this heuristic on various data sizes of instances. The results show that DC programming is well adapted to the structure of this problem, and that we have proposed an efficient decomposition. Every instance could be solved in less than half an hour. In practice, this enables the potential users to run the algorithm repeatedly to test various economical scenarios. At each iteration, the linear problem is solved very quickly and recovering feasibility does not pose any problem. We show the efficiency of the DC heuristic compared to using a commercial solver, especially for large instances, as the computing time is reduced from 75% to 90% for most of the largest instances. The degradation of the objective function is generally less than 1%. However, for the most difficult instances, we observe a significant optimality gap of about 3%. For further research, we propose to establish a more robust solution method, combining the DC approach with an efficient relaxation technique.

Appendix A. Mathematical model

Data sets:

$T(t \in T)$	planning horizon
$S(s \in S)$	set of suppliers
$\mathcal{I}(i \in \mathcal{I})$	set of plants
$\mathcal{J}(j \in \mathcal{J})$	set of distribution centres
\mathcal{JP}	set of private distribution centres
\mathcal{JH}	set of public distribution centres
$C(c \in C)$	set of customers
$\mathcal{O}(o \in \mathcal{O})$	set of capacity options
$\mathcal{P}(p \in \mathcal{P})$	set of products
\mathcal{P}_r	set of raw materials
\mathcal{P}_m	set of manufactured products
\mathcal{P}_f	set of finished products
Γ_i^+	set of successors of a node i
Γ_i^-	set of predecessors of a node i

For plants and distribution centres, we define two subsets denoted by subscripts c (I_c, JP_c, JH_c) and o (I_o, JP_o, JH_o) for the set of facilities that may close and open, respectively.

Decision variables:

Binary variables:

$x_i^t = 1$	if the facility i is active at t , 0 otherwise ($i \in S \cup \mathcal{I} \cup \mathcal{J}$)
$y_{i,o}^t = 1$	if the capacity option o is added to i at t , 0 otherwise ($i \in \mathcal{I} \cup \mathcal{J}$)
$z_{s,p}^t = 1$	if the supplier s is selected for p at t , 0 otherwise
$v_s^t = 1$	if the supplier s gives a discount at t , 0 otherwise

Continuous variables:

$f_{p,i,j}^t$	quantity of product p transferred from node i to node j at t
$k_{p,i}^t$	quantity of product p subcontracted for plant i at t
$g_{p,i}^t$	quantity of product p produced in plant i at t
$h_{p,j}^t$	quantity of product p held in distribution centre j at the beginning of t

Parameters:

All these parameters are part of the datasets and never modified by the heuristic.

Suppliers:

$HD_{s,p}^t = 1$	if the raw material p is available from s at t , 0 otherwise
$\bar{A}_{s,p}^t$	available capacity of supplier s for p at t

Plants:

\underline{K}_i	initial capacity of production at i
\bar{K}_i	maximal installable production capacity at i
$HA_{i,o}^t = 1$	if we can add the option o to facility i at t , 0 otherwise
KT_o	capacity of option o
\underline{U}_i	minimal percentage of utilisation of facility i (independent of t)
\bar{U}_i^t	maximal percentage of utilisation of facility i at t
$HP_{i,p}^t = 1$	if the plant i can manufacture p at t , 0 otherwise

Distribution centres:

The distribution centres have almost the same characteristics as the plants. The only distinctions are the following:

\underline{S}_j	initial storage capacity at j
KS_o	storage capacity of option o (if o is added to a distribution centre)
$HS_{j,p}^t = 1$	if the distribution centre j can store p at t , 0 otherwise

Customers:

$D_{c,p}^t$	known demand of customer c for product p at t
-------------	---

Products:

HC_p	1 if the product p can be subcontracted
$B_{p,p}$	quantity of p necessary to manufacture one unit of p (bill of materials)
$LT_{p,i}$	workload for the treatment of a unit p at facility $i \in \mathcal{I} \cup \mathcal{J}$

$LS_{p,j}$ workload for the storage of a unit p at distribution centre $j \in \mathcal{J}$
 $A_{i,j}$ number of deliveries from plant i to distribution centre j in one period

Costs:

CF_s fixed cost for selecting a supplier
 CD_s discount if at least two raw materials are supplied from s
 CO_i fixed cost for opening a facility i at a potential location
 CC_i fixed cost for closing an existing facility i
 $CA_{i,o}$ fixed cost for adding capacity option o to a facility i
 $CU_{i,0}$ fixed cost for operating a facility i
 $CU_{i,o}$ fixed cost for operating capacity option o at facility i
 $CP_{p,i}$ treatment cost of a unit p at facility $i \in \mathcal{I} \cup \mathcal{J}$
 $CP_{p,0}$ subcontracting cost of a unit of p
 $CS_{p,j}$ holding cost of a unit of p at distribution centre $j \in \mathcal{J}$
 $CT_{p,i,j}$ transportation cost of a unit of p from i to j

Objective function:

$$\begin{aligned} \text{Min } z = & \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}} CF_s \cdot x_s^t - CD_s \cdot v_s^t \quad (z_1 \text{ supplier selection fixed costs}) \\ & + \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}_0 \cup \mathcal{J} \mathcal{P}_o} CO_i \cdot (x_i^{t+1} - x_i^t) \quad (z_2 \text{ opening facility fixed costs}) \\ & + \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}_c \cup \mathcal{J} \mathcal{P}_c} CC_i \cdot (x_i^t - x_i^{t+1}) \quad (z_3 \text{ closing facility fixed costs}) \\ & + \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}_c \cup \mathcal{J} \mathcal{P}_c} \sum_{o \in \mathcal{O}} CA_{i,o} \cdot (y_{i,o}^{t+1} - y_{i,o}^t) \quad (z_4 \text{ adding capacity option costs}) \\ & + \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I} \cup \mathcal{J} \mathcal{P}} CU_{i,0} \cdot x_i^t + \sum_{o \in \mathcal{O}} CU_{i,o} \cdot y_{i,o}^t \quad (z_5 \text{ operating facility fixed costs}) \\ & + \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}_m} \sum_{i \in \mathcal{I}} CP_{p,i} \cdot g_{p,i}^t + CP_{p,0} \cdot k_{p,i}^t \quad (z_6 \text{ production variable costs}) \\ & + \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}_j} \sum_{j \in \mathcal{J}} CS_{p,j}^t \cdot \left(h_{p,j}^t + \sum_{i \in \mathcal{I}_j} \frac{1}{A_{i,j}} \cdot \frac{f_{p,i,j}^t}{2} \right) \quad (z_7 \text{ storage variable costs}) \\ & + \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}_j} \sum_{j \in \mathcal{J}} \sum_{c \in \mathcal{C}} CP_{p,j,c}^t \cdot f_{p,j,c}^t \quad (z_8 \text{ distribution variable costs}) \\ & + \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}_i} \sum_{i \in \mathcal{S} \cup \mathcal{I} \cup \mathcal{J}} CT_{p,i,j}^t \cdot f_{p,i,j}^t \quad (z_9 \text{ transportation variable costs}) \end{aligned}$$

The components z_1 – z_5 are fixed costs. The company has to pay the supplier selection fixed costs z_1 for each period and for each selected supplier. Whenever possible, the discount CD_s is removed from the cost. Other costs that are paid once a year are operating fixed costs. Other fixed costs are paid only once during the whole planning horizon, for example the opening costs z_2 , closing costs z_3 and capacity option costs z_4 . Indeed, these costs are considered as an investment for a new site (opening and adding capacity option costs) or as a cost for removing installed things and for paying for dismissing people.

The second part of the objective function consists of variable costs that depend on the product flows in the network. The production costs z_6 are the sum of the product costs for internal plants and the subcontracting costs. Storage variable costs z_7 are calculated as a function of the average inventory level which, in our model, is defined as half of the delivered quantity (from plants). This is calculated by the quotient of the total transported quantity divided by the number of deliveries in one period. The distribution cost z_8 is calculated as a function of the delivered quantity from distribution centres to customers. Finally, the transportation costs z_9 are related to the product flows between each node in the network.

Constraints

The following constraints are applied for all $t \in \mathcal{T}$:

$$\sum_{j \in \mathcal{I}_c} f_{p,j,c}^t \geq D_{c,p}^t \quad \forall c \in \mathcal{C}, \forall p \in \mathcal{P}_f \quad (1)$$

$$h_{p,j}^{t-1} + \sum_{i \in \mathcal{I}_j^-} f_{p,i,j}^t = \sum_{c \in \mathcal{I}_j^+} f_{p,j,c}^t + h_{p,j}^t \quad \forall j \in \mathcal{J}, \forall p \in \mathcal{P}_f \quad (2)$$

$$\sum_{i \in \mathcal{I}_i^-} f_{p,i,i}^t + k_{p,i}^t \geq \sum_{p \in \mathcal{P}_m} B_{p',p} \cdot g_{p,i}^t \quad \forall i \in \mathcal{I}, \forall p' \in \mathcal{P}_r \cup \mathcal{P}_m \quad (3)$$

$$g_{p,i}^t + k_{p,i}^t \geq \sum_{j \in \mathcal{I}_i^+} f_{p,i,j}^t \quad \forall i \in \mathcal{I}, \forall p \in \mathcal{P}_m \quad (4)$$

$$\sum_{p \in \mathcal{P}_m} LT_{p,i} \cdot g_{p,i}^t \geq \underline{U}_i \cdot \left(\underline{K}_i \cdot x_i^t + \sum_{o \in \mathcal{O}} KT_o \cdot y_{i,o}^t \right) \quad \forall i \in \mathcal{I} \quad (5)$$

$$\sum_{p \in \mathcal{P}_m} LT_{p,i} \cdot g_{p,i}^t \leq \overline{U}_i \cdot \left(\underline{K}_i \cdot x_i^t + \sum_{o \in \mathcal{O}} KT_o \cdot y_{i,o}^t \right) \quad \forall i \in \mathcal{I} \quad (6)$$

$$\sum_{p \in \mathcal{P}_j} \sum_{c \in \mathcal{I}_j^+} LT_{p,j} \cdot f_{p,j,c}^t \geq \underline{U}_j \cdot \left(\underline{K}_j \cdot x_j^t + \sum_{o \in \mathcal{O}} KT_o \cdot y_{j,o}^t \right) \quad \forall j \in \mathcal{J} \quad (7)$$

$$\sum_{p \in \mathcal{P}_j} \sum_{c \in \mathcal{I}_j^+} LT_{p,j} \cdot f_{p,j,c}^t \leq \overline{U}_j \cdot \left(\underline{K}_j \cdot x_j^t + \sum_{o \in \mathcal{O}} KT_o \cdot y_{j,o}^t \right) \quad \forall j \in \mathcal{J} \quad (8)$$

$$\sum_{p \in \mathcal{P}_f} LS_{p,j} \cdot \left(h_{p,j}^t + \sum_{i \in \mathcal{I}_j^-} \frac{1}{A_{i,j}} \cdot f_{p,i,j}^t \right) \leq \underline{S}_j \cdot x_j^t + \sum_{o \in \mathcal{O}} KS_o \cdot y_{j,o}^t \quad \forall j \in \mathcal{J} \quad (9)$$

$$\underline{K}_i \cdot x_i^t + \sum_{o \in \mathcal{O}} KT_o \cdot y_{i,o}^t \leq \overline{K}_i \quad \forall i \in \mathcal{I} \cup \mathcal{J} \quad (10)$$

$$\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} \overline{K}_i^t \cdot HP_{i,p}^t \cdot x_i^t \geq \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}} D_{c,p}^t \quad \forall p \in \mathcal{P}_f \quad (11)$$

$$\sum_{j \in \mathcal{J}} \overline{K}_j \cdot HS_{j,p}^t \cdot x_j^t \geq \sum_{c \in \mathcal{C}} D_{c,p}^t \quad \forall p \in \mathcal{P}_f \quad (12)$$

$$\sum_{i \in \mathcal{I}_s^+} f_{p,s,i}^t \leq z_{s,p}^t \cdot \overline{A}_{s,p}^t \quad \forall s \in \mathcal{S}, \forall p \in \mathcal{P}_r \quad (13)$$

$$z_{s,p}^t \leq x_s^t \cdot HD_{s,p}^t \quad \forall s \in \mathcal{S}, \forall p \in \mathcal{P}_r \quad (14)$$

$$v_s^t \leq \left(1/2 \sum_{p \in \mathcal{P}_r} z_{s,p}^t \right) \quad \forall s \in \mathcal{S} \quad (15)$$

$$\sum_{i \in \mathcal{I}_s^+} f_{p,s,i}^t \geq \alpha \cdot \overline{A}_{s,p}^t \cdot z_{s,p}^t \quad \forall s \in \mathcal{S}, \forall p \in \mathcal{P}_r \quad (16)$$

$$f_{p,i,i}^t \leq HP_{i,p}^t \cdot \overline{K}_i \quad \forall i \in \mathcal{I}, \forall i' \in \mathcal{I}_i^+, \forall p \in \mathcal{P}_m \quad (17)$$

$$f_{p,j,c}^t \leq HS_{j,p}^t \cdot \overline{K}_j \quad \forall j \in \mathcal{J}, \forall c \in \mathcal{I}_j^+, \forall p \in \mathcal{P}_f \quad (18)$$

$$k_{p,i}^t \leq x_i^t \cdot HCp \cdot M \quad \forall i \in \mathcal{I}, \forall p \in \mathcal{P}_m \quad (19)$$

$$y_{i,o}^t \leq x_i^t \cdot HA_{i,o}^t \quad \forall i \in \mathcal{I} \cup \mathcal{J} \mathcal{P}, \forall o \in \mathcal{O} \quad (20)$$

$$x_i^t \geq x_i^{t+1} \quad \forall i \in \mathcal{I}_c \cup \mathcal{J} \mathcal{P}_c \quad (21)$$

$$x_i^t \leq x_i^{t+1} \quad \forall i \in \mathcal{I}_o \cup \mathcal{J} \mathcal{P}_o \quad (22)$$

$$-1 \leq 2 \cdot x_j^t - x_j^{t-1} - x_j^{t+1} \leq 1 \quad \forall j \in \mathcal{J} \mathcal{H} \quad (23)$$

$$y_{i,o}^t \leq y_{i,o}^{t+1} \quad \forall i \in \mathcal{I} \cup \mathcal{JP}, o \in \mathcal{O} \tag{24}$$

$$x_i^t \geq y_{i,o}^{t-1} \quad \forall i \in \mathcal{I} \cup \mathcal{JP}, o \in \mathcal{O} \tag{25}$$

$$x_m^t, y_{i,o}^t, z_{s,p}^t, v_s^t \in \{0,1\} \quad \forall m \in \mathcal{I} \cup \mathcal{J} \cup \mathcal{S}, \forall i \in \mathcal{I} \cup \mathcal{J}, o \in \mathcal{O}, \forall s \in \mathcal{S}, \forall p \in \mathcal{P}_r \tag{26}$$

$$f_{p,m,n}^t, k_{p,i}^t, g_{p,i}^t, h_{p,j}^t \geq 0 \quad \forall m \in \mathcal{S} \cup \mathcal{I} \cup \mathcal{J}, \forall n \in \Gamma_m^+, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall p \in \mathcal{P}_m, \forall p' \in \mathcal{P}_f \tag{27}$$

The first four constraints (1)–(4) concern demand satisfaction for each layer of the network. The constraints (1) guarantee that all customer demands are met. The constraints (2) represent the flow conservation at distribution centres. The constraints (3) ensure that plants receive enough raw materials and semi-finished products in order to manufacture the required quantity of finished products. The constraints (4) require that the quantity of manufactured and subcontracted products at a plant is larger than its delivered quantity.

The constraints (5)–(10) concern the capacity (initial + options) of the plants and distribution centres. We suppose that a facility cannot run under its minimum level of utilisation and cannot exceed the maximum level of utilisation of its installed capacity. The constraints (5), (6) represent these conditions for the plants. The constraints (7), (8) are similar but applied to the distribution centres. Moreover, distribution centres cannot store more than their storage capacity (9). Finally, the installed capacity at any facility must not exceed its maximal installable capacity (10).

The constraints (11) and (12) are valid inequalities that strengthen the linear relaxation of the original problem. The constraints (13) require that suppliers deliver a raw material if, and only if, they are selected for this raw material. Moreover, their delivery cannot exceed their capacity. The constraints (14) state that a supplier, whenever selected, can only deliver a subset of raw materials. We suppose that a supplier offers a discount if it is selected for at least two raw materials (15). The constraints (16) propose a minimal amount of the delivered quantity from each supplier in order to avoid small artificial orders for obtaining the discount. This threshold α is arbitrary and has to be adjusted according to the context (0.1 or 0.2 are typical values). The constraints (17) and (18) state that a plant/distribution centre can only deliver products that it can manufacture/store. The constraints (19) require that only opened plants can have subcontracted products. The constraints (20) ensure that only an opened facility can add available capacity options. The constraints (21) and (22) prevent the facilities from changing their status (open or closed) more than once. Closed facilities cannot re-open, and open facilities cannot be closed. The status of public distribution centres cannot be changed after at least two periods (23). Additionally, capacity options cannot be removed (24). Finally, a facility where capacity options are added cannot be closed (25).

Appendix B. Calculation of the sequence Y^k

Let us consider some generic binary variable x_i and recall the penalty function $p(x) = \tau_i \sum_i \min\{x_i, 1-x_i\}$. Thus

$$\partial p(x_i) = \begin{cases} \tau_i & \text{if } x_i < 0.5 \\ -\tau_i & \text{if } x_i > 0.5 \\ [-\tau_i, \tau_i] & \text{if } x_i = 0.5 \end{cases}$$

For $x_i = 0.5$, we choose $\partial p(x_i) = 0$. In this way, we do not influence the variables that lie exactly in the middle of their bounds. This calculation is applied to variables $x_s^t, x_i^t, x_j^t, y_{i,o}^t, u_{s,p}^t$ and v_s^t .

Table 8
Summary of Y^k values / gradient of function z .

Variables	Value of $\partial(z)$
x_s^t	CS_s
x_i^t	$-CO_i + CU_{i,0}$ if $t=0$
$i \in \mathcal{I}_o$	$CO_i + CU_{i,0}$ if $t = T$
	$CU_{i,0}$ if $0 < t < T$
x_j^t	$-CO_j + CU_{j,0}$ if $t=0$
$j \in \mathcal{JP}_o$	$CO_j + CU_{j,0}$ if $t = T$
	$CU_{j,0}$ if $0 < t < T$
x_i^t	$CC_i + CU_{i,0}$ if $t=0$
$i \in \mathcal{I}_c$	$-CC_i + CU_{i,0}$ if $t = T$
	$CU_{i,0}$ if $0 < t < T$
x_j^t	$CC_j + CU_{j,0}$ if $t=0$
$j \in \mathcal{JP}_c$	$-CC_j + CU_{j,0}$ if $t = T$
	$CU_{j,0}$ if $0 < t < T$
$y_{i,o}^t$	$CA_{i,o} + CU_{i,o}$ if $t=0$
	$-CA_{i,o} + CU_{i,o}$ if $t = T$
	$CU_{i,o}$ if $0 < t < T$
	0
$z_{s,p}^t$	0
	0
v_s^t	$-CS'_s$

Table 9
Summary of Y^k values/continuous variables.

Variables	Value of $\partial(z)$
$f_{p,i}^t$	$CP_{p,i}$
$\bar{f}_{p,i}^t$	$CP_{p,o}$
$s_{p,j}^t$	$CS_{p,j}$
$q_{p,s,i}^t$	$CT_{s,i}$
$q_{p,i,i}^t$	$CT_{i,i}$
$q_{p,i,j}^t$	$CT_{i,j} + CS_{p,j} \cdot (0.5/L_{i,j})$
$q_{p,i,c}^t$	$CT_{i,c}$
$q_{p,j,c}^t$	$CT_{j,c} + CS'_{p,j}$

The function z is linear. The components of its derivative with respect to binary and continuous variables are detailed in Tables 8 and 9, respectively.

References

Achterberg, T., Berthold, T., 2007. Improving the feasibility pump. *Discrete Optimization* 4 (1), 77–86.

Beamon, B.M., 1998. Supply chain design and analysis: models and methods. *International Journal of Production Economics* 55 (3), 281–294.

Benders, J., van Nunen, J.A.E.E., 1983. A property of assignment type mixed integer linear programming problems. *Operations Research Letters* 2, 47–52.

Bertacco, L., Fischetti, M., Lodi, A., 2007. A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization* 4, 63–76.

Cordeau, J.-F., Pasin, F., Solomon, M.M., 2006. An integrated model for logistics network design. *Annals of Operations Research* 144 (1), 59–82.

Cortinhal, M.J., Captivo, M.E.V., 2003. Upper and lower bounds for the single source capacitated location problem. *European Journal of Operational Research* 151 (2), 333–351.

Daskin, M.S., Snyder, L.V., Berger, R., 2005. Facility location in supply chain design. In: Langevin, A., Riopel, D. (Eds.), *Logistics Systems: Design and Optimization*. Springer, US, pp. 39–65.

Drezner, Z., Nickel, S., 2009. Constructing a DC decomposition for ordered median problems. *Journal of Global Optimization* 45 (2), 187–201.

Fischetti, M., Glover, F., Lodi, A., 2005. The feasibility pump. *Mathematical Programming* 104 (1), 91–104.

French, A.P., Wilson, J.M., 2007. An lp-based heuristic procedure for the generalized assignment problem with special ordered sets. *Computers & Operations Research* 34 (8), 2359–2369.

Hardin, J.R., Nemhauser, G.L., Savelsbergh, M.W., 2007. Analysis of bounds for a capacitated single-item lot-sizing problem. *Computers & Operations Research* 34 (6), 1721–1743 part Special Issue: Odysseus 2003 Second International Workshop on Freight Transportation Logistics.

- Le Thi, H.A., Nguyen Trong, P., Pham, D.T., 2007. A continuous dc programming approach to the strategic supply chain design problem from qualified partner set. *European Journal of Operational Research* 183 (3), 1001–1012.
- Le Thi, H.A., Pham, D.T., 2005. The DC (difference of convex functions) programming and DCA revisited with dc models of real world nonconvex optimization problems. *Annals of Operations Research* 133 (1–4), 23–46.
- Le Thi, H.A., Pham, D.T., 2008. A continuous approach for the concave cost supply problem via DC programming and DCA. *Discrete Applied Mathematics* 156 (3), 325–338.
- Le Thi, H.A., Pham, D.T., Le, D., 1999. Exact penalty in d.c. programming. *Vietnam Journal of Mathematics* 27 (2), 169–178.
- Levi, R., Shmoys, D.B., Swamy, C., 2004. Lp-based approximation algorithms for capacitated facility location. *Lecture Notes in Computer Science* 3064, 21–27.
- Melachrinoudis, E., Min, H., 2000. The dynamic relocation and phase-out of a hybrid, two-echelon plant/warehousing facility: a multiple objective approach. *European Journal of Operational Research* 123 (1), 1–15.
- Melkote, S., Daskin, M.S., 2001. Capacitated facility location/network design problems. *European Journal of Operational Research* 129 (3), 481–495.
- Melo, M.T., Nickel, S., Saldanha da Gama, F., 2006. Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. *Computers & Operations Research* 33 (1), 181–208.
- Melo, M.T., Nickel, S., Saldanha da Gama, F., 2009. Facility location and supply chain management—a review. *European Journal of Operational Research* 196 (2), 401–412.
- Owen, S.H., Daskin, M.S., 1998. Strategic facility location: a review. *European Journal of Operational Research* 111 (3), 423–447.
- Pham, D.T., El Bernoussi, S., 1986. Algorithms for solving a class of non convex optimization problems: methods of subgradients. In: Hiriart-Urruty, J.-B. (Ed.), *Fermat Days 85: Mathematics for Optimization*, vol. 129. North-Holland Mathematics Studies, Elsevier Science Publishers, Amsterdam, pp. 249–270.
- Sahin, G., Süral, H., 2007. A review of hierarchical facility location models. *Computers & Operations Research* 34 (8), 2310–2331.
- Thanh, P.N., 2008. Conception et planification stratégique des réseaux logistiques. Ph.D. Thesis, University of Nantes.
- Thanh, P.N., Bostel, N., Péton, O., 2008. A dynamic model for facility location in the design of complex supply chains. *International Journal of Production Economics* 113 (2), 678–693.
- Thanh, P.N., Bostel, N., Péton, O., 2010. A linear relaxation-based heuristic for logistics network design. *Computers & Industrial Engineering* 59 (4), 964–975.
- Toland, J., 1978. Duality in non convex optimization. *Journal of Mathematical Analysis and Applications* 66, 399–415.

A New Consistent Vehicle Routing Problem for the Transportation of People with Disabilities

Dominique Feillet and Thierry Garaix

Ecole des Mines de Saint-Etienne, CMP Georges Charpak, F-13541, Gardanne, France

Fabien Lehuédé and Olivier Péton

IRCCyN, Ecole des Mines de Nantes, 4 rue Alfred Kastler, F-44307, Nantes, France

Dominique Quadri

Université d'Avignon et des Pays du Vaucluse, Laboratoire Informatique d'Avignon, F-84911, Avignon, France

In this article, we address a problem of the transportation of people with disabilities where customers are served on an almost daily basis and expect some consistency in the service. We introduce an original model for the time-consistency of the service, based on so-called time-classes. We then define a new multiday vehicle routing problem (VRP) that we call the Time-Consistent VRP. We address the solution of this new problem with a large neighborhood search heuristic. Each iteration of the heuristic requires solving a complex VRP with multiple time windows and no waiting time which we tackle with a heuristic branch-and-price method. Computational tests are conducted on benchmark sets and modified real-life instances. Results demonstrate the efficiency of the method and highlight the impact of time-consistency on travel costs. © 2013 Wiley Periodicals, Inc. NETWORKS, Vol. 000(00), 000–000 2013

Keywords: consistent vehicle routing problem; transportation of people with disabilities; multiple time windows; heuristic branch-and-price

1. INTRODUCTION

Among the diverse categories of problems that have been investigated in operations research, vehicle routing problems (VRPs) probably constitute one of the most emblematic class. In practice, their purpose is generally to propose optimized

distribution plans while delivering goods to customers (e.g., be they individuals, firms, or retailers). In most of the many models developed for the routing of vehicles, a single period of time is considered (say, a day). This modeling, however, corresponds to two very different practical situations: either the routes are intended to be performed once and recomputed with new data the next day, or the routes are repeated identically every single day over a long time horizon. However, there is an intermediate situation in which demands are almost identical from day to day. In addition to the usual cost minimization objective, a new criterion might then make sense: limiting the variability of the routes between the different periods of the time horizon. The problem then becomes a multiday VRP where routes are expected to show some regularity from day to day.

An important illustration of this problem was given by Groër et al. [14] and concerns fast parcel delivery (involving big companies such as UPS). The ability to assure consistent schedules and to limit the number of drivers that a customer has to deal with can significantly improve the satisfaction of these customers: facilitating receipt operations, improving trust between the delivery companies and their customers, and so forth. A major advantage can thus be provided in comparison with competing companies.

Another interesting case, which motivated this article, concerns the transportation of people with mental disabilities from and to day care centers [19]. Because of their lack of autonomy, most of the persons concerned have no vehicle and are not capable of traveling by public transport. Consequently, medicosocial centers usually resort to specialized transportation companies to organize and perform the daily trips. Customers need to be transported regularly but sometimes with some small variations during the week. For example, a person may not ask for transportation or may not have the same address every day. In addition, these users

Received March 2013; accepted August 2013

Correspondence to: D. Feillet and T. Garaix; e-mail: feillet.garaix@emse.fr

Contract grant sponsor: GdR RO (Operations Research Group of the Scientific Research National Center)

DOI 10.1002/net.21538

Published online in Wiley Online Library (wileyonlinelibrary.com).

© 2013 Wiley Periodicals, Inc.

are particularly sensitive to changes. The issue of keeping consistent time schedules is therefore fundamental.

In this article, we investigate how consistency can be considered in VRPs. Because the expectations of stability are context-dependent, as will be detailed below, we essentially address the transportation of people with disabilities and the stability of the time of services. Because there is a paucity of works on this subject, we first discuss the notion and the modeling of consistency, before introducing a new model (section 2). We then introduce the time-consistent VRP (TCVRP) and propose a solution approach based on a large neighborhood search (LNS) (section 3). Computational experiments evaluating the behavior of this algorithm are then described (section 4), before the article is concluded (section 5).

2. MODELING THE TIME-CONSISTENCY OF ONE CUSTOMER

The issue of constructing consistent vehicle routes is related to many important lines of research in operations research such as robustness or stochasticity [3, 13]. However, consistency is generally sought in relation to unexpected events that disrupt the transport.

In contrast, we consider here the consistency of services with respect to deterministic data with slight variations from day to day, which has rarely been investigated in a routing context. Our study mainly compares with the recent paper of Groër et al. [14] which introduces the consistent VRP. Since its first online publication, this problem has been investigated in [18, 22, 27]. The consistent vehicle problem clearly shares some similarities with the courier delivery problem [26] which, in addition, considers time windows, uncertain service times, uncapacitated vehicles and that frequent customers should not necessarily be served by a single vehicle. A few other studies show some connections with this subject [1, 2, 28]. Finally, the consistent VRP has some relationship with multiperiod VRPs [12], where the assignment of visit days is also considered.

From the customer point of view, consistency improves the quality of service. Having consistent schedules helps the customer to get prepared for the service, to manage material flows, to schedule upstream and downstream tasks, and so forth. Assigning the same driver to the different visits of a customer facilitates the delivery process and strengthens the relationship between suppliers and customers [6, 16, 24]. Such criteria can play an essential role in the choice of a supplier [5].

For drivers, having consistent sequences of visits also presents many advantages. It helps develop a good knowledge of the usual itineraries and traffic conditions, thus enabling drivers to better estimate distribution times and potential delays, reducing transportation times [6, 15, 24, 25]. Driver familiarity with the customer also helps in applying security and administrative procedures [16]. With all these improvements, the transportation process is generally considered more comfortable by the drivers [25].

In the remainder of the article, we essentially restrict the study to the consistency of schedules. Our assumption is that once a time-consistent solution is computed, finding a reasonably consistent assignment of drivers should be possible. This assumption is checked experimentally in section 4. The issue of simultaneously optimizing time- and driver-consistency is left for future research.

To discuss the modeling of time-consistency, let us consider a customer i , a time horizon $D = \{1, \dots, |D|\}$, and times of service (or schedule) $W = (w_1, \dots, w_{|D|})$ for customer i over this horizon: the service of customer i starts at time w_d on day d . For the sake of simplicity, note that we assume here that customer i is visited every day of the time horizon. Note also that W is assumed to be known. The method to optimize W will be the subject of section 3.

We first review the model proposed by Groër et al. [14] and then describe our model which generalizes the former.

Note that we do not consider explicitly customers with several addresses. Actually, time-consistency is important for a given address. A person with two distinct addresses is modeled as two people (one per address).

2.1. Modeling Time-Consistency Through Time Slacks

The time-consistency for customer i is ensured in [14] by limiting the difference between the latest and earliest service dates. More formally, a time of service variation is measured as:

$$TC_{\text{slack}}(W) = \max_{1 \leq d \leq |D|} w_d - \min_{1 \leq d \leq |D|} w_d \quad (1)$$

This definition is relevant in the context of the small package shipping industry. Figure 1 illustrates for five different schedules W_1, \dots, W_5 , the measure of time-consistency following Groër et al. for a horizon ranging from Monday to Friday. F1

However, considering other application contexts, the preceding modeling needs to be questioned. A first remark is that it is particularly suitable for small variations in service times. Thus, schedules $W_1 = (8:00, 8:03, 8:05, 8:00, 8:02)$ and $W_2 = (8:00, 8:00, 8:05, 8:05, 8:00)$, such that $TC_{\text{slack}}(W_1) = TC_{\text{slack}}(W_2) = 5$, can be considered equivalent in terms of time-consistency. However, it is less clear for $W_3 = (8:00, 9:00, 8:20, 8:30, 8:40)$ and $W_4 = (8:00, 8:02, 8:05, 8:58, 9:00)$. Although $TC_{\text{slack}}(W_3) = TC_{\text{slack}}(W_4) = 60$, W_4 shows much more regularity, with two sets of almost identical times of service around 8:00 and 9:00.

Following this observation, a second important remark is that while W_1 or W_2 undoubtedly define more consistent schedules than W_4 , should the two schedules W_4 and $W_5 = (8:00, 8:02, 8:05, 9:58, 10:00)$, with $TC_{\text{slack}}(W_4) = 60$ and $TC_{\text{slack}}(W_5) = 120$ respectively, be considered so different? In the following section and in the remainder of this article, we introduce a new modeling framework for time-consistency which assumes that such schedules are equivalent.

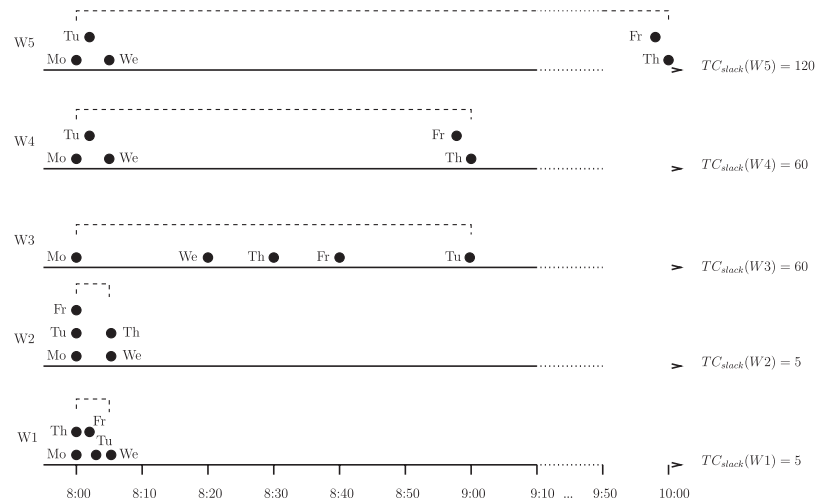


FIG. 1. Time-consistency measure of Groër et al. [14].

As could be expected, Groër et al. [14] attain a very high level of consistency in their solutions, thus justifying their modeling. Two factors can be considered determinant in this behavior. First, vehicles deliver 100 to 150 customers everyday. Second, many of these customers are visited only once during the time horizon, thus giving many opportunities for substitutions in the routes. As a consequence, consistent schedules can be preserved for the relevant customers, giving rise to a strong time-consistency and mostly avoiding being concerned by the above remarks. From an algorithmic perspective, the chosen approach was to define template routes that respect the so-called precedence principle: “If customers i and j are both served by the same vehicle on a specific day and i is serviced before j , then customer i must receive service before customer j from the same vehicle on all days where they both require service.” Note that this principle also deals with driver-consistency.

For the transportation of people with disabilities, the situation is very different. Vehicles only visit a very small number of people, who make transportation requests almost everyday, thus rendering substitution between customers much more complicated and increasing the differences between the service times. This explains why we introduced the modeling developed below.

2.2. Modeling Consistency Through Time-Classes

From the above discussion, we introduce a new consistency measure that follows these principles: a small gap between service times is not significant; a large gap induces a disturbance, whatever the gap.

Let us consider the parameter $\delta > 0$ reflecting the sensitivity of the customers to schedule variations. To evaluate the time-consistency for customer i , we define the notion of time-classes as follows: the times of service w_d and $w_{d'}$ are allowed to be assigned to the same time-class if $|w_d - w_{d'}| \leq \delta$. Hence, the times of service $\{w_1, \dots, w_{|D|}\}$ can be divided into time-classes.

It can easily be seen from the above definition that there is no unique way of assigning a set of times of service to time-classes. Let us go back to the example of schedule $W_3 = (8:00, 9:00, 8:20, 8:30, 8:40)$. With parameter $\delta = 10$ one can, for example, construct time-classes $\{\text{Mo}\}, \{\text{We, Th}\}, \{\text{Fr}\}, \{\text{Tu}\}$ or $\{\{\text{Mo}\}, \{\text{We}\}, \{\text{Th, Fr}\}, \{\text{Tu}\}\}$.

We define the time-consistency $\text{TC}(W)$ of a schedule $W = (w_1, \dots, w_{|D|})$ as the minimal number of time-classes into which $w_1, \dots, w_{|D|}$ can be ranked:

$$\text{TC}(W) = \min C \quad (2)$$

subject to

$$|w_d - w_{d'}| > \delta \Rightarrow C_d \neq C_{d'} \quad (d, d' \in D), \quad (3)$$

$$C_d \leq C \quad (d \in D), \quad (4)$$

$$C_d \in \{1, \dots, |D|\} \quad (d \in D), \quad (5)$$

$$C \in \{1, \dots, |D|\}. \quad (6)$$

Constraints (3) impose the assignment of different classes for service times whose difference is greater than δ . Constraints (4) with objective function (2) compute the minimal number of classes required C .

Figure 2 outlines the proposed consistency through time-classes considering schedules W_1 to W_5 introduced in subsection 2.1, with parameter $\delta = 10$. Note that with this new measure, schedules W_1 and W_2 are still the most consistent, schedules W_4 and W_5 are considered equivalent, and schedule W_3 is strongly penalized. The method to compute values $\text{TC}(W)$ and to assign visits to time-classes is presented in the following subsection.

2.3. Computation of Time-Classes

It can easily be seen that model (2)–(6) defines a graph coloring problem. Thus, undirected graph $G = (D, E)$ could be introduced, where edge $(d, d') \in E$ if and only if $|w_d - w_{d'}| > \delta$. Computing the minimal number of time-classes for

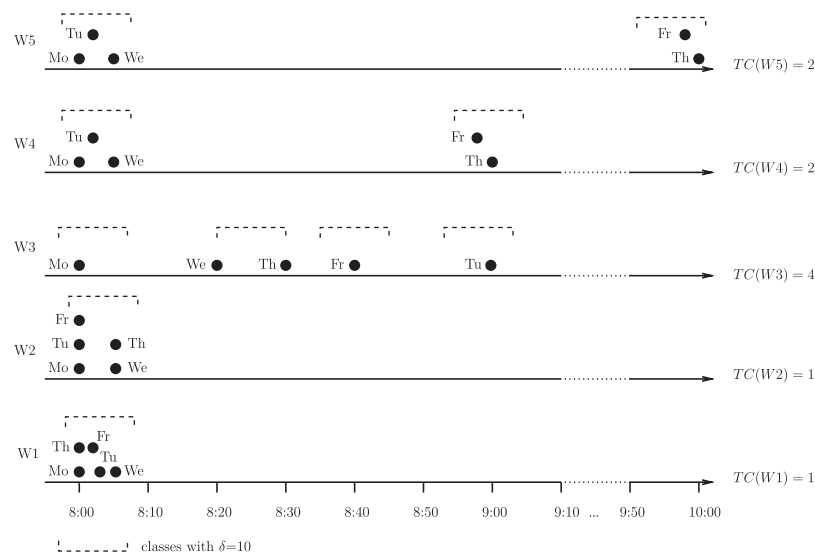


FIG. 2. Time-class consistency.

schedule W then amounts to coloring the vertices in G , such that two vertices connected with an edge are colored with a different color.

Although graph coloring problems are generally NP-hard, graph G exhibits a very special structure which implies that the problem is polynomial. This is highlighted by the subsequent equivalent definition of set E : edge $(d, d') \in E$ if and only if $[w_d - \delta/2, w_d + \delta/2] \cap [w_{d'} - \delta/2, w_{d'} + \delta/2] = \emptyset$. As such, the coloring problem can be solved using a greedy algorithm (Algorithm 1). In Algorithm 1, vertices are colored in the increasing order of times of service w_d , with the first available color. Let d represent the first vertex colored with the current color. A sufficient condition for a vertex to be colorable using the current color is to be compatible with vertex d .

The time-classes that would be constructed for schedules W_1 to W_5 with $\delta = 10$, applying Algorithm 1, can be observed in Fig. 2.

Algorithm 1 Computation of time-classes

```

 $\{d_1, \dots, d_{|D|}\} \leftarrow$  vertices  $\{1, \dots, |D|\}$  sorted in increasing
order of times of servicecolor vertex  $d_1$  with color 1
 $d \leftarrow d_1$ 
for  $i = 2$  to  $|D|$  do
  if edge  $(d, d_i) \notin E$  then
    color vertex  $d_i$  with the same color as  $d$ 
  else
    color vertex  $d_i$  with the next color
     $d \leftarrow d_i$ 
  end if
end for

```

3. THE TCVRP

In the remainder of the article, we first assess how the proposed definition of time-consistency can be handled in

a solution approach. Second, we evaluate the impact of considering time-consistency on vehicle routes.

The original application includes various specific constraints and criteria that can complicate the readability of the subsequent results. We thus introduce and address a simplified problem. This is called the TCVRP and is basically a multiday VRP with the additional constraint of optimizing time-consistency over days.

An important difference to be noted compared to previous sections is that customers are initially located at the depot and request to be transported back home with regular arrival times of services, with a set of vehicles all departing at the same time (time 0) from the depot. In the real-application that motivates this article, the people with mental disabilities all had to arrive at a given time and expected to be picked up at regular times. From a computational point of view, however, the two situations are completely symmetric and equivalent.

We first define the TCVRP in section 3.1. Then, the proposed solution algorithm is outlined in section 3.2. Finally, sections 3.3, 3.4, and 3.5 provide details on the different steps of the algorithm.

3.1. Definition of the TCVRP

Let us denote D the set of days in the time horizon and consider a complete graph $G = (V, A)$, where $V = \{0, \dots, n\}$ represents a depot (vertex 0) and the destinations of n customers (vertices 1 to n). Customers need to be transported from the depot to their destination on some days of the time horizon, defined by $p_i^d = 1$ if customer i has to be transported on day d , $p_i^d = 0$ otherwise. Set K represents a fleet of identical vehicles available at the depot every day of the time horizon. The capacity of a vehicle is limited to Q customers. A travel time $t_{ij} > 0$ is defined for every arc $(i, j) \in A$. This travel time corresponds equivalently, without loss of generality, to a travel cost. A parameter δ is given to represent the

maximal difference of times of service that can be ranked in the same time-class.

Two objectives are pursued. The first is to minimize travel costs, that is, the sum of the route costs throughout the horizon. The second is to maximize time-consistency, that is, to minimize the maximal number of time-classes over the n customers. The number of time-classes for a customer is defined in equations (2)–(6), where times of service are related to routing decisions.

Note that no time windows are defined for deliveries. In addition, waiting times are not allowed. Without this constraint, waiting could improve time-consistency.

The TCVRP then consists in optimizing these two objectives by finding a set of at most $|K|$ routes every day of the time horizon D such that (i) all the requested deliveries are performed, (ii) all routes satisfy vehicle capacities. Seeing that D is assumed to remain quite small (typically a week), the problem can rather be stated as the problem of solving $|D|$ monoobjective subproblems, where the maximal number of time-classes is limited and successively defined as $C = 1, \dots, |D|$. The model (7)–(25) presented below shows a mathematical formulation for one of these monoobjective subproblems, with a limited number of time-classes, namely at most C time-classes per customer. Note, however, that the solution method and experiments presented later deal with the complete (multiobjective) TCVRP. The model is as follows:

$$\text{Min } z = \sum_{d \in D} \sum_{k \in K} \sum_{(i,j) \in A} t_{ij} x_{ijk}^d \quad (7)$$

s.t.

$$\sum_{k \in K} \sum_{j \in V} x_{ijk}^d = p_i^d \quad (i \in V \setminus \{0\}, d \in D), \quad (8)$$

$$\sum_{j \in V} x_{jik}^d - \sum_{j \in V} x_{ijk}^d = 0 \quad (i \in V \setminus \{0\}, k \in K, d \in D), \quad (9)$$

$$\sum_{i \in V \setminus \{0\}} x_{0ik}^d = 1 \quad (k \in K, d \in D), \quad (10)$$

$$\sum_{i \in V \setminus \{0\}} x_{i0k}^d = 1 \quad (k \in K, d \in D), \quad (11)$$

$$\sum_{i \in V \setminus \{0\}} \sum_{j \in V} x_{ijk}^d \leq Q \quad (k \in K, d \in D) \quad (12)$$

$$w_{jk}^d \leq w_{ik}^d + t_{ij} + (1 - x_{ijk}^d)M \quad ((i,j) \in A, k \in K, d \in D), \quad (13)$$

$$w_{jk}^d \geq w_{ik}^d + t_{ij} - (1 - x_{ijk}^d)M \quad ((i,j) \in A, k \in K, d \in D), \quad (14)$$

$$w_i^d \leq w_{ik}^d + (1 - \sum_{j \in V} x_{ijk}^d)M \quad (i \in V \setminus \{0\}, k \in K, d \in D), \quad (15)$$

$$w_i^d \geq w_{ik}^d - (1 - \sum_{j \in V} x_{ijk}^d)M \quad (i \in V \setminus \{0\}, k \in K, d \in D), \quad (16)$$

$$w_{0k}^d = 0 \quad (k \in K, d \in D), \quad (17)$$

$$w_i^{d_2} - w_i^{d_1} \leq \delta + M y_i^{d_1 d_2} \quad (i \in V \setminus \{0\}, d_1, d_2 \in D), \quad (18)$$

$$c_i^{d_1} + 1 \leq c_i^{d_2} + M(1 - y_i^{d_1 d_2}) \quad (i \in V \setminus \{0\}, d_1, d_2 \in D), \quad (19)$$

$$y_i^{d_1 d_2} \in \{0, 1\} \quad (i \in V \setminus \{0\}, d_1, d_2 \in D), \quad (20)$$

$$c_i^d \leq C \quad (i \in V, d \in D), \quad (21)$$

$$x_{ijk}^d \in \{0, 1\} \quad ((i,j) \in A, k \in K, d \in D), \quad (22)$$

$$w_{ik}^d \geq 0 \quad (i \in V \setminus \{0\}, k \in K, d \in D), \quad (23)$$

$$w_i^d \geq 0 \quad (i \in V \setminus \{0\}, d \in D), \quad (24)$$

$$c_i^d \in \{1, \dots, C\} \quad (i \in V, d \in D). \quad (25)$$

The decision variables have the following interpretation:

- x_{ijk}^d : 1 if arc (i, j) is used by vehicle k on day d , 0 otherwise,
- w_i^d : time of service for customer i on day d ,
- w_{ik}^d : time of service for customer i on day d if it is carried out by vehicle k ,
- c_i^d : number of the time-class assigned to the delivery of customer i on day d .

The intermediate binary variables $y_i^{d_1 d_2}$ are set to 1 if the times of service of customer i on days d_1 and d_2 are increasing in this order and they are not allowed to belong to the same time-class ($w_i^{d_2} - w_i^{d_1} > \delta$), and 0 otherwise.

Constraints (8) state that customer i is serviced on day d if and only if he or she needs to be transported on that day. Constraints (9) are classical flow conservation constraints. Constraints (10) and (11) enable vehicles to leave the depot and return to the depot once a day at most, respectively. Constraints (12) present vehicle capacities. Constraints (13) and (14) define time variables. They linearize the logical expression

$$\forall (i,j) \in A, \forall k \in K, \forall d \in D, x_{ijk}^d = 1 \Rightarrow w_{jk}^d = w_{ik}^d + t_{ij}.$$

Note that vehicles are not allowed to wait to improve time-consistency. Constraints (15) and (16) define times of service for customers, independently of the vehicle used. They linearize the logical expression

$$\forall i \in V \setminus \{0\}, \forall k \in K, \forall d \in D, \sum_{j \in V} x_{ijk}^d = 1 \Rightarrow w_i^d = w_{ik}^d.$$

Constraints (17) set vehicle starting times at the depot. For every customer, constraints (18)–(20) enforce the assignment of distinct time-classes to service times that differ from more than δ . They linearize the logical expression

$$\forall i \in V \setminus \{0\}, \forall d_1, d_2 \in D, w_i^{d_2} - w_i^{d_1} > \delta \Rightarrow c_i^{d_1} + 1 \leq c_i^{d_2}.$$

Constraints (21) limit the maximum number of time-classes allowed for every customer.

Finally, we note that in the case of $C \geq |D|$, constraints (18)–(20), (21), and (25) can be removed since every day can be solved independently. Furthermore, the case $C = 1$ can also be simplified by replacing those constraints by

$$|w_i^{d_2} - w_i^{d_1}| \leq \delta \quad (i \in V \setminus \{0\}, d_1, d_2 \in D). \quad (26)$$

3.2. Solution of the TCVRP with a LNS Approach

In this section, we describe the framework of the algorithm proposed for the solution of the TCVRP. Subsequent sections explain the different procedures introduced here.

Time-consistency constraints significantly complicate the problem as they introduce temporal dependences between routes. Very few VRPs in the literature exhibit similar features. The most closely related problems we were able to find are those with synchronization constraints [4, 17]. As pointed out by Rousseau et al. [21], synchronization constraints considerably limit the applicability of traditional local search techniques. Assessing the feasibility or impact of a very small move can be a large and complex problem. From this observation, we decided to address the solution of the TCVRP with a LNS approach.

The LNS heuristic, first introduced by Shaw [23] has been successfully applied to VRPs in recent years [8, 20]. Although our implementation differs from the initial algorithm of Shaw, the proposed algorithm follows the fundamental principle of LNS: the heuristic starts from a feasible solution and alternates between the destruction of a large part of the solution and the reconstruction of the resulting partial solution.

The main principle of the proposed heuristic is to start from a solution that is good from the cost point of view and to reduce the number of time-classes progressively, recording the best found solution for each level of time-consistency (thus solving the bi-objective version of the TCVRP). To diversify the search, a perturbation mechanism and a restart procedure are performed periodically.

The initial solution is computed by considering each day independently and solving (heuristically) the corresponding VRPs. The solution obtained tends to minimize the total distance traveled during the time horizon. As time-consistency is completely ignored, it is likely that some customers will have a large number of time-classes.

At every iteration, the destruction phase selects one day for which all routes are destroyed. The reconstruction phase relies on the definition of time constraints that enable the maximal number of time-classes for the customers to be controlled. Our approach is to select a customer whom we attempt to reduce the number of time-classes and to limit the number of time-classes for other customers, to varying degrees depending on their distance from the current maximal number of time-classes. These constraints can be expressed in the form of multiple time windows associated with customers. The problem to solve is then a variant of the VRP that we call

the VRP with multiple time windows and no-wait constraints (VRPmTW-nw). This problem is tackled with a heuristic branch-and-price solution method.

Algorithm 2 outlines the main steps of the LNS heuristic.

Algorithm 2 General scheme of the LNS algorithm

```

1:  $sol \leftarrow$  initial feasible solution
2:  $D' \leftarrow D$ 
3: repeat
4:   every  $p_1$  useless iterations without improvement:
     diversify or restart
5:   select a day  $d$  at random in  $D'$ 
6:   select a customer  $i$  with a maximum number of time-
     classes among those served on day  $d$ 
7:   define an instance of the VRPmTW-nw for day  $d$ 
     according to  $sol$  and  $i$ 
8:    $s \leftarrow$  solution of the VRPmTW-nw or null if infeasible
9:   if  $s \neq \mathbf{null}$  then
10:     $sol \leftarrow$  replace routes of day  $d$  in  $sol$  by the routes
      of  $s$ 
11:  end if
12:  if  $D' \setminus \{d\} \neq \emptyset$  then  $D' \leftarrow D' \setminus \{d\}$  else  $D' \leftarrow D$ 
13: until a computing time limit is met

```

The LNS algorithm is initialized (line 1) by solving the VRP defined every day. For this purpose, we apply the heuristic branch-and-price algorithm that is used for reconstruction (see section 3.4), initialized with routes obtained by applying the Clarke and Wright algorithm. In this case, no time windows are defined for the customers. The main loop (Lines 3 to 13) alternates between the selection of a day to destroy and the reconstruction phase.

Day selection is done at random (line 5), ensuring, however, that all days have been selected before one is destroyed again (line 12). In line 6, a customer is selected to focus time-class reduction on this customer. The one chosen has a maximal number of time-classes among customers served on that day. From the customers who satisfy this criterion, we select one for whom the time-class involved on that day has the lowest cardinality. The purpose is to give more chances of deleting that time-class in future iterations.

Following the selection of a day and a customer, an instance of the VRPmTW-nw is defined (line 7). The way this instance is constructed is presented in section 3.3. The heuristic branch-and-price method used for the solution of the VRPmTW-nw (line 8) is detailed in section 3.4. When possible, the current solution sol is then updated (line 10) and the LNS iterations are continued until the computing time limit is reached.

When p_1 useless iterations of the LNS operators have not led to any improvement of one of the best known solutions (one per level of time-consistency), diversification or restart components are called for (line 4). These two components and the selection process between them are presented in section

3.5. Iterations are counted as useless when they do not succeed in reducing the number of customers with a maximal number of time-classes.

3.3. Time-Window Definition for the VRPmTW-nw

The definition of the instance of the VRPmTW-nw to be solved at every iteration of the algorithm is based on the current solution sol , the day d that was deconstructed and the selected customer i . As explained in section 3.2, the objective of the deconstruction and reconstruction phases is to reduce the number of time-classes for i and to limit the number of time-classes for other customers according to a set of rules that are described in this section.

To describe our scheme, we introduce the following notation. The number of time-classes for $j \in V \setminus \{0\}$ in sol is denoted $TC_j^{sol}(D)$. The number of time-classes for j after the deconstruction of day d is denoted $TC_j^{sol}(D \setminus \{d\})$. $TC^{sol}(D)$ is the maximal number of time-classes of the customers during horizon D , that is the time-consistency value for sol . Finally, the construction of the VRPmTW-nw instance relies on a parameter denoted $classMax_j$. This represents the maximal number of time-classes allowed for $j \in V \setminus \{0\}$ in the new solution to be constructed.

According to this notation, $classMax_j = TC_j^{sol}(D) + 1$ means that any time of service is possible for customer j on day d (no time constraint). On the contrary, $classMax_j = TC_j^{sol}(D)$ generally means that the time of service for customer j on day d has to be compatible with the times of service on other days for this customer. The sole exception is met when $TC_j^{sol}(D \setminus \{d\}) < TC_j^{sol}(D)$, that is when the time of service of customer j on day d in sol was defining a time-class on its own.

In the above paragraph, compatible means that the time of service for customer j on day d can be integrated into one of the existing time-classes for this customer. Given a time-class whose lowest time of service is t_{inf} and highest time of service is t_{sup} , the condition for the time of service w_j^d to be compatible with this time-class is $t_{sup} - \delta \leq w_j^d \leq t_{inf} + \delta$. This condition can be expressed in the form of a time window condition: $w_j^d \in [t_{sup} - \delta, t_{inf} + \delta]$. Being compatible with the different time-classes then comes down to satisfying multiple time windows.

The number and the size of time windows (can be none) assigned to a customer in the travel cost minimization problem on day d depends on how the $classMax$ parameter has been defined. Let us now describe the proposed definition of $classMax$. We recall that i is the customer selected at the current iteration of the LNS algorithm, for whom a decrease in the number of time-classes is expected.

1. Four cases are considered for customers $j \in V \setminus \{0\}, j \neq i$:
 - (a) if $TC_j^{sol}(D) = TC^{sol}(D)$:
 - i. if this condition held in the past p_2 iterations, $classMax_j = TC_j^{sol}(D) + 1$,

- ii. else, if $TC_j^{sol}(D) = TC^{sol}(D)$ for no more than p_2 iterations, $classMax_j = TC_j^{sol}(D)$,
- (b) if $TC_j^{sol}(D) < TC^{sol}(D)$:
 - i. $classMax_j = TC_j^{sol}(D) + 1$ with probability $\min(p_3(TC^{sol}(D) - TC_j^{sol}(D)), 1)$,
 - ii. otherwise $classMax_j = TC_j^{sol}(D)$.
2. For customer i , if $TC^{sol}(D) > 1$, $classMax_i = TC^{sol}(D) - 1$, otherwise $classMax_i = 2$.

Rules 1(a)i and 1(b)i state that the number of time-classes for customer j can be increased, whereas it remains the same in rules 1(a)ii and 1(b)ii. According to $classMax_j$, time windows are then defined for j . Two situations are possible:

- $classMax_j = TC_j^{sol}(D \setminus \{d\}) \Rightarrow$ [multiple time windows],
- $classMax_j > TC_j^{sol}(D \setminus \{d\}) \Rightarrow$ [no time window].

The rationale behind using probability $\min(p_3(TC^{sol}(D) - TC_j^{sol}(D)), 1)$ in rules 1(b)i and 1(b)ii is that customers with a large number of time-classes are given less opportunity to increase this number. Parameter p_3 is set to 0 at the beginning of the algorithm and increased by 0.1 at every restart (see section 3.5) with a maximum value of 0.5.

The only possibility that the number of time-classes in the new solution exceeds $TC^{sol}(D)$ comes from rules 1(a)i and 2 (second case), where the purpose is to inject some diversification.

Note finally that the definition of time windows may not actually reduce the number of time-classes of i in the first case of rule 2. In fact, when $TC_i^{sol}(D \setminus \{d\}) = TC_i^{sol}(D) = TC^{sol}(D)$, no solution with $TC^{sol}(D) - 1$ time-classes exists (remember that i is selected from the customers with a maximal number of time-classes on day d). In this case, $classMax_i$ will not exactly represent the maximal number of time-classes for customer i . Definition $classMax_i = TC^{sol}(D) - 1$ will rather be handled by defining one time window per time-class of customer i except for the time-class in which the time of service of customer i on day d was ranked. The purpose is to decrease the size of this time-class and eventually to empty it in further iterations.

Example The preceding principles are illustrated in the following example: we consider a solution for 15 customers on 5 days with $\delta = 5$, which results in a time-consistency $TC^{sol}(D) = 3$. The selected customer i is customer 7 and day 2 is deconstructed. Columns 2 to 6 of Table 1 present the time schedule of the solution for the 15 customers. Columns $TC_j^{sol}(D)$ and $TC_j^{sol}(D \setminus \{d\})$ present the number of time-classes of each customer in the current and the deconstructed solutions, respectively. The following columns then present the rule applied to each customer (with an increased number of time-classes allowed for customers 6 and 8) and the time windows defined according to these values (“-” stands for no time window).

T1

3.4. Solution of the VRPmTW-nw

The problem to solve at each iteration of the reconstruction phase of the LNS consists in finding a set of routes with

TABLE 1. Definition of multiple time windows for a solution with 15 customers, 5 days and $\delta = 5$.

Customers	day 1	day 2	day 3	day 4	day 5	$TC_j^{sol}(D)$	$TC_j^{sol}(D \setminus \{d\})$	Rule	classMax	Time windows for day 2
1	54.6	54.6	91.8	91.1	91.1	2	2	1(b)ii	2	[44.6;64.6] [81.8;101.1]
2	12.7				29.4	2	2			
3	23	69			19.1	2	1	1(b)ii	2	–
4				55.2	55.2	1	1			
5	53.9			7.7		2	2			
6	71.8	15.6	23.8	21.8	21.8	2	2	1(b)i	3	–
i=7	37.3	94.3	118.3	32.2	117.6	3	2		2	[27.3;42.2] [108.3;127.6]
8	40.8	40.8	78	77.3	77.3	2	2	1(b)i	3	–
9	2.4	2.4	2.4	44.4	2.4	2	2	1(b)ii	2	[0;12.4] [34.4;54.4]
10	84.6		10.6	8.6	8.6	2	2			
11		84.8				1	0	1(b)ii	1	–
12		29.8	38		51.8	2	2	1(b)ii	2	[28;48] [41.8;61.8]
13	26.6	26.6	63.8	63.1	63.1	2	2	1(b)ii	2	[16.6;36.6] [53.8;73.1]
14		108.6		17.9		2	1	1(b)ii	2	–
15			8.1			1	1			

minimal cost that serve all customers involved on that day. One additional condition, compared to the standard VRP is that deliveries are restricted by multiple time windows. Also, no waiting time is allowed. We call this problem VRPmTW-nw and describe here the solution approach developed for it. Although the VRPmTW-nw is new (to the best of our knowledge), a heuristic branch-and-price solution scheme could relatively easily be adapted from existing VRPTW solution schemes.

The principle of the branch-and-price algorithm for the VRPTW [7, 9] is: a set-covering formulation is defined where path variables represent the set of feasible routes. Due to the very large number of possible routes, the formulation is first restricted to a limited subset of variables (restricted master problem) and then extended using a pricing subproblem. At each iteration of the so-called column generation scheme, the linear relaxation of the restricted master problem is solved and the pricing problem is called for, given a vector of optimal dual variables. The aim of the pricing problem is then to determine one or several feasible routes of negative reduced cost. By subtracting the values of dual variables from arc costs, the pricing problem can be expressed as an elementary shortest path problem with resource constraints and solved with dynamic programming. This whole scheme is embedded in a branch-and-price algorithm so as to obtain optimal integer solutions. Branching rules consist in imposing or forbidding arcs in the solution.

Regarding the VRPmTW-nw, most of the previous approaches can be applied in a similar way. The only difference lies in the pricing problem and concerns the feasibility of routes. While extending paths in the dynamic programming algorithm, one has to address multiple time windows and forbidden waiting times. Feasibility is assessed by simply checking that the destination node is reached within one of its time windows. As waiting times are not allowed, the condition that two paths arrive at the destination at exactly the same time is added to the dominance rule. In fact, earlier arrival, which is usually preferred as it gives more opportunity for further extensions, could here prevent some customers from

being reached. The consequence is a significantly weakened dominance.

In the process of maintaining elementary paths and reinforcing the dominance rule, label definition includes the computation of the set of reachable nodes [10]. Here, reachable nodes are those that were not visited in the partial path represented by the label and whose last time window ends after the earliest possible arrival.

Actually, in view of the way multiple time windows are defined (see section 3.3), it is possible that a large proportion of customers are not subject to time windows. So as to counterbalance the negative effect of the no-waiting-time condition on the dominance rule, we implement the following mechanism. Although computing the set of reachable nodes for a label, we determine that we do not fall into the situation where all the remaining nodes have no time windows. For these labels, the traditional dominance rule is applied.

Through these adaptations, the VRPmTW-nw can theoretically be solved exactly. As the branch-and-price algorithm has to be called many times using relatively similar data, a single pool of columns is kept and enriched throughout the successive calls of the algorithm. A filtering algorithm is performed at each new call to consider only feasible columns (regarding the current data) in the model. However, the need to solve a VRPmTW-nw instance at every iteration of Algorithm 2 also implies very limited computing times. Hence, a heuristic variant of the previous branch-and-price scheme is developed.

When solving the pricing problem with dynamic programming, the set of labels conserved with every node of the graph is limited to the $p_4 = 20$ best nondominated labels, in order of increasing time. Exploration in the dynamic programming algorithm is also limited with a limited discrepancy search policy [11]. Basically, extensions are limited to the $p_5 = 2$ nearest neighbors (sorted according to the reduced cost). However, extensions to other neighbors are permitted with the limit of at most $p_6 = 2$ extensions for a complete route.

It is well-known that column generation is subject to slow convergence. To avoid long series of column generation iterations without a clear improvement in the objective value, we stop after a given number of iterations with improvements less than $p_7 = 1\%$. This number of iterations is set to $p_8 = 5$ when a feasible solution has already been found, $p_9 = 10$ otherwise.

Finally, a simple diving strategy is implemented for the branching phase. When the current optimal solution of the relaxed restricted master problem is fractional, instead of splitting the search into two descendant nodes, we only explore the descendant node where an arc is imposed.

These three heuristic mechanisms avoid, respectively, long pricing problem solution times, a large number of iterations of column generation at each node of the search tree, and a large number of nodes in the search tree, thus ensuring that many instances of the VRPmTW-nw can be solved in a reasonable time. A drawback of these mechanisms is that they involve a relatively large set of parameters (p_4 to p_9). The value of the different parameters defined above was set through preliminary experiments, which demonstrated, however, that the method was not very sensitive to these values.

3.5. Restart and Diversification

The diversification operator works as follows. The first step is to select a customer with a maximal number of time-classes for a maximal number of iterations. This customer is then removed from the solution and reinserted every requested day as the first customer of a route; more precisely, the route that yields the minimal insertion cost among the existing routes on each day. Since vehicles always start from the depot at time 0, the time of service for this customer will always be the same, leading to only one time-class for this customer. This simple operator thus enforces consistency for a customer who was stuck with a large number of time-classes for a long time and starts the search in a completely new direction.

The restart operator replaces diversification when a solution with a time-consistency equal to 1 has been found since the last diversification or restart. It consists in restoring one of the best known solutions as the current solution. Our choice is to restore solutions with any potential number of time-classes. Thus, the first call to the restoration operator will generate a solution with $|D|$ time-classes, then $|D| - 1$, and so forth. When value 1 is reached, the process starts again from $|D|$. As the LNS algorithm tends to search for solutions with a small time-consistency, this restart operator aims to spend some time improving best known solutions with larger time-consistencies.

4. COMPUTATIONAL EXPERIMENTS

The LNS heuristic was coded in C++ and run on a machine with a 2.5-GHz processor and 3 GB of RAM and evaluated on two types of data instances. First, we considered the so-called

small instances of [14], which comprise 10 instances with 10 or 12 customers over a 3-day horizon. The average level of presence of the customers is around 70%. The instances are named *conVRP* x - y , with $x = 10$ or 12 for the number of customers and $y = 1, \dots, 5$, for the instance index. In these instances, a route duration limit is included, which can easily be handled in the model and the algorithm.

Then, we considered 14 real cases arising from day care centers for mentally disabled children or adults in the area of Nantes (France). The centers have to transport between 15 and 65 people everyday (inbound trips in the morning and outbound trips in the evening). The planning horizon is 5 days. For each original instance, the demand of the transported persons is represented by vectors of five binary values: 1 if the person has to be transported on that day and 0 otherwise. We modified the original demand vector so that the level of presence varied from $\rho = 0.5$ to $\rho = 0.9$ in steps of 0.1. To obtain a level of presence ρ with realistic data, we made the following assumptions: a proportion ρ^2 of the people make a demand everyday; a proportion $(1 - \rho^2)$ of the people have a level of presence of $\frac{\rho}{1+\rho}$; for the latter category, the level of presence on Tuesdays, Wednesdays, and Thursdays is twice that of Mondays and Fridays. The resulting 70 instances are named data z - x , where $z = 10 \times \rho$ and x is the number of customers.

The original instances [19] include additional constraints such as individual time windows or maximum trip duration, several possible addresses during the week, incompatibilities between people, wheelchair constraints, a heterogeneous fleet, and so forth. We ignored these complicating constraints to focus on the consistency aspect.

As explained in the introductory section of the article, a post processing phase is carried out to assign drivers to routes in such a way that driver-consistency is optimized. The objective is to minimize the average number of drivers per customer. This is introduced and discussed in [24], where the authors explain that it is representative of customer satisfaction in terms of driver-consistency. They also propose several integer programming formulations. In our computations, we used their (DC) model which minimizes the average number of drivers transporting each customer.

4.1. Small Instances

The aim of this subsection is threefold: (i) to show that the problem is difficult to solve in practice (i.e., the limits of the integer linear model); (ii) to validate the proposed heuristic in comparison with the best solutions found; and (iii) to compare the proposed model with Groër et al. [14].

4.1.1. Comparison with the Integer Linear Program

Table 2 first evaluates the ability of the integer linear programming formulation introduced in section 3.1 to solve the problem. As it seems almost incapable of solving any of the instances, we temporarily relax vehicle capacity constraints, maximum route length constraints, and assume unitary demands. These instances, derived from the original

T2

TABLE 2. Optimal (ILP) and heuristic (LNS) solutions on modified small instances ($\delta = 5$).

Time-classes:	ILP			Avg LNS		
	3	2	1	3	2	1
RconVRP10-1	92.91	92.91	92.91	92.91	92.91	92.91
RconVRP10-2	80.42	80.42	–	80.42	80.42	80.99
RconVRP10-3	94.12	94.12	94.37	94.12	94.12	94.37
RconVRP10-4	93.71	93.71	–	93.71	93.71	94.09
RconVRP10-5	83.84	83.84	–	83.84	84.54	96.08
RconVRP12-1	103.65	103.65	–	103.65	103.65	104.40
RconVRP12-2	73.89	73.89	–	73.89	73.89	81.27
RconVRP12-3	82.77	82.77	–	82.77	82.77	83.12
RconVRP12-4	97.57	97.57	–	97.57	97.57	101.91
RconVRP12-5	83.63	83.63	–	83.63	83.63	89.25

ones, are named “RconVRP x - y .” The sensitivity parameter δ is set to 5. Three runs of 10 h using ILOG-CPLEX 12.0 are performed on each instance, setting the maximal number of time-classes C – in inequality (21) – to 3, 2 and 1, respectively. The optimal solutions found are presented in the “ILP” section of the table. Optimal solutions are compared to the solutions provided by our heuristic in 5 min of computing time (each run provides a solution for every number of time-classes): the “Avg LNS” part presents the average results over 10 runs of the heuristic.

The ILP formulation was, in most cases, unable to solve the 1-time-class problems to optimality within 10 h. Moreover, ILOG-CPLEX was not able to improve the lower bounds given by the 2-time-classes problems. All the known optimal solutions were reached at each run of the LNS algorithm, except for instance RconVRP10-5; for this one, our algorithm was not able to find the optimal solution with 2 time-classes; instead solutions of costs 84.5 and 84.69 were found, 8 and 2 times, respectively, for an average value of 84.54.

T3 To evaluate the results obtained with 1 time-class, Table 3 provides the average and best solutions found by LNS (after the 10 runs), the best feasible solution built by the ILP, and the best known solution among these solutions (“Best”). Values that match the best known solutions are written in bold. Our heuristic failed to find the best known solution for only two

TABLE 3. 1-time-class solutions on modified small instances ($\delta = 5$).

	Avg LNS	Best LNS	ILP	Best
RconVRP10-1	92.91	92.91	92.91	92.91
RconVRP10-2	80.99	80.99	82.83	80.99
RconVRP10-3	94.37	94.37	94.37	94.37
RconVRP10-4	94.09	94.09	94.53	94.09
RconVRP10-5	96.08	96.01	96.01	96.01
RconVRP12-1	104.40	104.40	106.87	104.40
RconVRP12-2	81.27	81.25	81.25	81.25
RconVRP12-3	83.12	83.12	83.12	83.12
RconVRP12-4	101.91	101.91	101.91	101.91
RconVRP12-5	89.25	89.25	89.25	89.25

TABLE 4. Heuristic solutions and driver-consistency on small instances ($\delta = 5$).

Instance	Opt. cost	LNS cost	Gap	Number of customers with		
				1 driver	2 drivers	3 drivers
conVRP10-1	122.03	118.70	2.8%	7	3	0
conVRP10-2	99.07	96.57	2.6%	8	2	0
conVRP10-3	123.41	123.16	0.2%	9	1	0
conVRP10-4	126.89	126.89	0.0%	10	0	0
conVRP10-5	109.31	104.04	5.1%	7	3	0
conVRP12-1	144.02	143.23	0.6%	10	2	0
conVRP12-2	89.54	83.27	7.5%	9	3	0
conVRP12-3	119.69	113.74	5.2%	9	3	0
conVRP12-4	141.37	137.26	3.0%	10	2	0
conVRP12-5	117.42	110.17	6.6%	9	3	0

cases. For the RconVRP10-5 instance, it found a suboptimal solution of cost 96.7 once out of 10 runs. For the RconVRP12-2 instance, our heuristic returned a 1-time-class solution with cost 81.34 instead of 81.25 twice out of 10 runs.

For three instances out of 10, our heuristic found better results than the ILP formulation. Moreover, since the gap between the best-LNS solutions and the lower bounds—taken from the 2-time-classes optimal solutions of Table 2—is 3.8% on average and 14.5% at maximum, we can be confident in the quality of the solutions computed by our heuristic on the modified small instances.

4.1.2. Comparison with the Approach of Groër et al.

Table 4 presents a comparison between our approach and that proposed by Groër et al. [14]. For this set of experiments, the original instances *conVRP x - y* introduced in [14] are considered.

The purpose here is not to compare the solution methods but rather to discuss the modeling differences. Indeed, Groër et al. enforce in their model that a single driver is used for every customer, while we do not consider this issue implicitly but rather optimize the number of drivers *a posteriori*. Also, Groër et al. do not introduce the concept of time-classes but they consider the maximal difference between the service times. Their heuristic minimizes (implicitly) that difference while it is bounded in their ILP formulation. The chosen limit is 5. This situation corresponds to the set of 1-time-class solutions provided with our heuristic, with parameter δ set to 5.

The optimal costs obtained in [14] are reported in the column “Opt.” of Table 4.¹ The costs of the 1-time-class solutions obtained for these instances with our heuristic are reported in the next column, followed by the observed gap with “Opt.” The difference in cost is not significant on average (3.4%), showing improvements of up to 7.5% when relaxing driver consistency. The impact on driver-consistency can be evaluated by the last three columns of the table, where we report the number of customers with

¹ Optimal values given in [14] include service durations. They are discarded in the costs of Table 4.

TABLE 5. Solutions of the heuristic method on real instances ($\delta = 10$).

C_{max}	5	4	3	2	1
data5-15	700.8	664.7	663.2	670.5	705.7
data5-21	827.1	775.8	775.9	776.9	823.7
data5-25	724.5	617.8	617.8	618	655.9
data5-26	845.2	777.3	771.1	778.5	829.2
data5-27	1010	946.5	940.8	947.2	1066.8
data5-32		1002.1	987.8	988.4	1023.2
data5-41	1465.5	1439.5	1467.8	1467.1	1615.1
data5-44	1176.5	1158.6	1150.4	1162	1219.2
data5-46	1482	1474.3	1474.2	1497.3	1562.1
data5-48		1460.1	1457.7	1462	1590.6
data5-55		1622.7	1605.2	1606.8	1855.6
data5-59	2804.4	2787.8	2778.6	2789.9	3059.8
data5-64	2141.1	2140.1	2172.3	2163	2321.2
data5-65	1795.5	1797.4	1798.1	1802.3	2052.7
data6-15	736.4	689.7	688.3	691.4	731.3
data6-21	849.3	794.7	793.6	795.1	812.9
data6-25	788.9	695.4	683.1	682.8	736.1
data6-26	853.3	843.8	839.4	841.1	888.1
data6-27	1040.6	951.1	952.9	954.9	1030.9
data6-32	1004.2	1002	992.4	992.4	1013.5
data6-41	1883	1521.2	1540.2	1565.8	1841
data6-44	1351.5	1251.3	1246.6	1251.1	1340.9
data6-46	1618.7	1506.1	1503.1	1515.7	1608.1
data6-48	1529.7	1533.2	1536	1529.6	1663.7
data6-55	1895	1890.1	1858.8	1861.4	2011.1
data6-59	3026.4	3023	3028.4	3017.7	3337.3
data6-64	2338	2327.8	2325.4	2333.7	2558.1
data6-65	2038.4	2018.4	2003.3	2012.8	2104.7
data7-15	766	749.6	746.9	751.3	773.6
data7-21	846.5	832.3	831.6	833.6	880.7
data7-25	776.3	726.4	724.5	727.4	754.9
data7-26	900.5	885.4	882.6	890.2	918.9
data7-27	1122.8	1059.4	1057	1059.9	1108.6
data7-32	1249.5	1103	1094.6	1101.8	1146.8
data7-41	1746.5	1724.5	1669.4	1664.1	1800.1
data7-44		1306.2	1301.4	1301.8	1369.2
data7-46	1697.8	1679	1659.5	1664.1	1717.1
data7-48	1729	1723.3	1713.7	1716.1	1854
data7-55	2003.8	1945.8	1931.8	1933.8	2038.3
data7-59	3388.7	3364.2	3340.2	3348.1	3667.7
data7-64		2608.6	2630.7	2612	2817.7
data7-65		2235.4	2261.6	2280.4	2380.2

one, two, and three drivers, obtained after optimally assigning drivers to the routes in the heuristic solutions. Here, perfect driver-consistency is only reached for instance “conVRP10-4.” However, the proportion of customers with two drivers never exceeds 30% and no customer has three drivers on these instances. The average losses in driver-consistency (counted in percentage of customers served by more than one driver) equal to 1.8 and 2.6% on instances with 10 and 12 customers, respectively, enable average cost savings of 2.1 and 4.6%. Hence, given the cost reduction obtained, allowing some flexibility in driver-consistency can be worthwhile when possible.

4.2. Real Instances

In this subsection, we investigate the efficiency of our solution approach on the 70 real instances. Remember that in

these instances no route duration limit is considered. We first present a set of results (benchmark results) obtained with a computing time of 1 h per instance. The objective here is to give some reference results for further research. Reference results are also provided to evaluate the efficiency of our heuristic with more limited computing times and assess the impact of time-consistency on routing costs.

4.2.1. Benchmark Results Tables 5 and 6 detail the results obtained by a single run of our heuristic, limited to 1 h, on real instances with a time-consistency parameter δ set to 10. T5,6

Note that the values given in the first column of this table are the best solutions found with exactly the number of time-classes indicated in the label of the column. In practice, it is clear that a result with fewer time-classes and smaller routing costs would be preferred. For example, on instance data5-15, solutions with 5 and 4 time-classes are actually dominated by the solution with 3 time-classes. These kinds of solutions are generally generated because of diversification and are not really optimized. We decided to present the results as such considering that “non-dominated” solutions can easily be deduced. Statistics on these results and also on “non-dominated” solutions are proposed in Table 7. The first part of the table (Rough results) summarizes the results given in Table 5 and provides the average value for the instances for which a solution was found. In the second part of the table T7

TABLE 6. Solutions of the heuristic method on real instances ($\delta = 10$) continued.

C_{max}	5	4	3	2	1
data8-15	817.5	776.5	775.3	777.2	792.8
data8-21	910.4	899.2	899.2	902.9	916.9
data8-25		860.7	853.9	853.3	857
data8-26	1110.6	974.6	965	968.5	1000.8
data8-27	1341.1	1198.8	1186.2	1186.2	1247.3
data8-32	1172.2	1159.8	1152.4	1154.6	1175.4
data8-41		1917.5	1906.9	1910.5	1989.7
data8-44		1433.5	1410.8	1412.9	1462.7
data8-46	1806.2	1799	1785	1782.7	1856.7
data8-48	1862.1	1843.8	1848.4	1848	1942.4
data8-55	2122.5	2085.1	2054.5	2065.3	2197.2
data8-59	3657.5	3656.7	3640.7	3645.5	3848
data8-64	2823	2792.8	2774.4	2809.6	2914.9
data8-65	2498.8	2504.2	2469	2461.6	2570.3
data9-15	861.5	796.1	797.4	799.1	816.1
data9-21	1044.8	1003.6	998.8	999.9	1008.9
data9-25		917	895.4	894.6	908.1
data9-26	1066.3	1025.4	1024.6	1024.6	1025.6
data9-27	1327.8	1221.9	1211.7	1213.7	1257.7
data9-32	1380.9	1210.5	1189.6	1193.1	1202.8
data9-41		2065.3	2052.4	2038.9	2040.4
data9-44	1622.6	1617.7	1562.4	1571.3	1671.2
data9-46		1855.2	1844.5	1857.4	1899.9
data9-48	2089.7	1993.2	1982.2	1982.2	2064.2
data9-55		2183.6	2184.5	2197	2484.2
data9-59	4033.1	4021	3936.9	3932.9	4084
data9-64	3030.3	3023.1	3004.4	2999.4	3037.5
data9-65		2656.7	2625.9	2626.8	2686.3

TABLE 7. Statistics on real instances

	Number of time-classes				
	5	4	3	2	1
Rough results					
Number of solutions	56	70	70	70	70
Average value	1584.51	1573.54	1564.72	1568.14	1660.65
Non-dominated solutions					
Number of solutions	70	70	70	70	70
Average value	1561.93	1561.95	1563.50	1568.14	1660.65
Average cost of consistency	–	0.00%	0.10%	0.30%	5.90%

(Non-dominated solutions), values of “dominated” solutions are replaced by values of solutions that “dominate” them (also, the best solution with fewer time-classes is considered when no solution was found). In the last line of the table, gaps indicate the increase in travel costs observed compared to solutions where 5 time-classes are allowed ($C = 5$).

Several comments can be drawn from Tables 5–7. The first rather surprising result is that feasible solutions with a small number of time-classes could be found for all 70 instances. Only some solutions with exactly 5 time-classes were not obtained on the whole set of instances. This latter point is not really surprising since having 5 different classes can sometimes counter-optimize the solution. For example, it is possible that the solution obtained by solving the VRP each day independently (which is optimal in terms of routing) does not introduce any customer with 5 time-classes. In this case, solutions with 5 time-classes are naturally dominated and not explored during the search.

In addition, we remark that imposing time-consistency does not penalize routing costs so much. As long as 2 time-classes are accepted, the impact on routing costs is less than 1%. When a limit of 1 time-class per customer is strictly enforced, routing costs are only increased by 5.90%, which is not negligible but might be offset by the increase in consistency for the customers.

Regarding the solution method, it is interesting to add that an average of 6,000 iterations could be performed in 1 h, thus allowing a large exploration of the solution space (remember that in one iteration, one fifth (1 day) of the solution can be completely redrawn).

T8,9

4.2.2. Efficiency of the Algorithm with Shorter Computing Times In Tables 8 and 9, we compare the number of solutions found and the values of the solutions, for computing

TABLE 8. Number of solutions found.

Computing time	Number of time-classes				
	5	4	3	2	1
60 s	29	68	67	61	44
600 s	43	70	70	70	65
1 h	56	70	70	70	70

TABLE 9. Gap compared to (non-dominated) solutions found in one hour of computing time.

Computing time	Number of time-classes				
	5	4	3	2	1
60 s	1.38%	1.45%	2.22 %	3.30 %	2.25 %
600 s	0.56%	0.58%	0.86%	1.19 %	0.56%

times varying between 60 s, 600 s, and 1 h, respectively. Table 8 is based on rough results and Table 9 considers non-dominated solutions. Note that when computing a gap, only instances for which a solution was found for both computing time limits are considered.

One can see from Tables 8 and 9 that “good” solutions can be obtained relatively quickly using the heuristic (less than 10 min). After these few minutes, the results are only marginally improved. The only exception concerns solutions with 1 time-class, which are the most difficult to compute and sometimes require many iterations to obtain. The shortest computing times are not assessed as, due to the complexity of the problem to be solved at each iteration of the algorithm, the method is not designed to be efficient in a few seconds.

4.2.3. Impact of Parameter δ Tables 10 and 11 aim to show how solutions are impacted when a tighter gap is fixed. T10,11

We compare the number of solutions found and the values of the solutions obtained when defining $\delta = 5$ with the previous results ($\delta = 10$). Table 10 is based on rough results, Table 11 considers non-dominated solutions. When computing a gap, only instances for which a solution was found for both values δ are considered.

The main observation that can be drawn from Table 10 is that solutions with one time-class can still be found most of the time (63 times out of 70). However, we cannot know whether the missing solutions could not be obtained because of our algorithm or they do not exist. Table 11 highlights that when more than 1 time-class is allowed, decreasing δ only slightly impacts the costs. Regarding solutions with 1 time-class, one can observe a clear increase in the costs.

4.2.4. Driver-Consistency Table 12 indicates the average number of drivers assigned to customers. Let us recall that T12

TABLE 10. Number of solutions found.

Number of time-classes	5	4	3	2	1
$\delta = 10$	56	70	70	70	70
$\delta = 5$	68	70	70	70	63

TABLE 11. Gap compared to (non-dominated) solutions found with $\delta = 10$.

Number of time-classes	5	4	3	2	1
$\delta = 5$	0.57%	0.66%	0.69%	1.29%	7.09%

TABLE 12. Average number of drivers per customer.

Number of time-classes	5	4	3	2	1
Average number of drivers	1.43	1.40	1.35	1.30	1.24

T13

driver-consistency is not explicitly addressed in our model and approach, but is rather handled in post processing so that the assignment of drivers to the routes computed with the heuristic optimizes the driver-consistency. Table 13 gives the percentage of customers having a given number of drivers out of all the customers of all the solutions (from 5 time-classes to 1) of the 70 instances. Note that these results are compiled from the solutions presented in Table 5, that is, the set of rough results.

The average number of drivers remains quite limited though not explicitly handled in the algorithm. In addition, we remark that the majority of customers have one or two drivers.

5. CONCLUSIONS

The content of this article was initially motivated by the optimization of vehicle routes when transporting people with disabilities to day care centers. So as to capture the specific consistency expectations in this context, a new time-consistency measure, based on time-classes, was defined. From this measure, we introduced the TCVRP and proposed a solution method based on large neighborhood search.

Experiments tend to demonstrate the great difficulty of this new problem and the efficiency of the heuristic proposed. An important result is that, at least for the (realistic) instances used for the computations, very consistent solutions could always be found with a relatively small impact on travel costs.

Though driver-consistency was not addressed in the model nor in the solution approach, a fairly good consistency could also be maintained for drivers.

A clear development of this work is to provide efficient exact algorithms to solve the TCVRP. The integer programming formulation proposed in this article is not solved efficiently by the solver so it would be worth investigating more evolved modeling, valid inequalities and branch-and-cut approaches. From the application perspective, introducing the diverse constraints that were not considered here would also be important. In addition, other possible models of time-consistency could be investigated: for example, it might be interesting to consider the number of time-class changes over successive periods, instead of just counting the number of classes.

TABLE 13. Percentage of customers having a given number of drivers.

Number of drivers	1	2	3	4	5
Percentage of customers	69.06%	25.9%	4.56%	0.46%	0.00%

Finally, we introduced another new routing problem in the paper, the so-called VRPmTW-nw. Given the inherent interest and difficulty of this problem, further research regarding its solution certainly deserves to be conducted.

REFERENCES

- [1] F. Baniel, *Prise en compte d'objectifs de stabilité pour l'optimisation de collecte de déchets*, Ph.d thesis, Université de Toulouse, France, 2009.
- [2] W.C. Benton and M.D. Rossetti, The vehicle scheduling problem with intermittent customer demands, *Comput Oper Res* 19 (1992), 521–531.
- [3] D.J. Bertsimas, A vehicle routing problem with stochastic demand, *Oper Res* 40 (1992), 574–585.
- [4] D. Bredström and M. Rönnqvist, Combined vehicle routing and scheduling with temporal precedence and synchronization constraints, *Eur J Oper Res* 191 (2008), 19–31.
- [5] T. Choi and J.L. Hartley, An exploration of supplier selection practices across the supply chain, *J Oper Manag* 14 (1996), 333–343.
- [6] J.M. Day, P.D. Wright, T. Schoenherr, M. Venkataramanan, and K. Gaudette, Improving routing and scheduling decisions at a distributor of industrial gases, *Omega* 37 (2009), 227–237.
- [7] G. Desaulniers, J. Desrosiers, and M.M. Solomon (Editors), *Column generation*, GERAD 25th anniversary series. Springer, US, 2005.
- [8] L.-M. Rousseau, E. Prescott-Gagnon, and G. Desaulniers, A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows, *Networks* 54 (2009), 190–204.
- [9] D. Feillet, A tutorial on column generation and branch-and-price for vehicle routing problems, *4OR–Q J Oper Res* 8 (2010), 407–424.
- [10] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen, An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems, *Networks* 44 (2004), 216–229.
- [11] D. Feillet, M. Gendreau, and L.-M. Rousseau, New refinements for the solution of vehicle routing problems with branch and price, *INFOR* 45 (2007), 239–256.
- [12] P.M. Francis, K.R. Smilowitz, and M. Tzur, “The period vehicle routing problem and its extensions,” *The vehicle routing problem: Latest advances and new challenges*, Vol. 43 of *Operations Research/Computer Science Interfaces Series*, B. Golden, S. Raghavan, E. Wasil, R. Sharda, and S. Voß, (Editors), Springer, US, 2008, pp. 73–102.
- [13] M. Gendreau, G. Laporte, and R. Seguin, A tabu search heuristic for the vehicle routing problem with stochastic demands and customers, *Oper Res* 44 (1996), 469–477.
- [14] C. Groër, B. Golden, and E. Wasil, The consistent vehicle routing problem, *Manuf Serv Oper Manag* 11 (2009), 630–643.
- [15] M.A. Haughton, Measuring and managing the learning requirements of route reoptimization on delivery vehicle drivers, *J Bus Logist* 23 (2001), 45–66.
- [16] M.A. Haughton, Assigning delivery routes to drivers under variable customer demands, *Transp Res Part E Logist Transp Rev* 43 (2007), 157–172.

- [17] I. Ioachim, J. Desrosiers, F. Soumis, and N. Bélanger, Fleet assignment and routing with schedule synchronization constraints, *Eur J Oper Res* 119 (1999), 75–90.
- [18] A.A. Kovacs, S.N. Parragh, and R.F. Hartl, A template based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks* 63 (2014), 60–81.
- [19] F. Lehuédé, C. Pavageau, and O. Péton, “Un système d’aide à la décision pour planifier les transports vers les établissements médico-sociaux,” *Proc Handicap 08 Conf*, Paris, June 2008.
- [20] S. Ropke and D. Pisinger, An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, *Transp Sci* 40 (2006), 455–472.
- [21] L.-M. Rousseau, M. Gendreau, and G. Pesant, The synchronized vehicle dispatching problem, Technical Report CRT-2003-11, Centre de Recherche sur les Transports, Université de Montréal, Canada, 2003.
- [22] H. Schneider, A. Reinholz, and H.-W. Graf. Integrated modeling and optimization of the consistent vehicle routing problem with the super node concept, *Proc 9th Metaheuristics Int Conf (MIC 2011)*, Udine, Italy, July 25–28, 2011.
- [23] P. Shaw, “Using constraint programming and local search methods to solve vehicle routing problems,” *Principles and practice of constraint programming – CP98*, Vol. 1520 of *Lecture notes in Computer Science*, M. Maher and J.-F. Puget (Editors), Springer, Berlin/Heidelberg, 1998, pp. 417–431.
- [24] K. Smilowitz, M. Nowak, and T. Jiang, Workforce management in periodic delivery operations, *Transp Sci* 47 (2013), 214–230.
- [25] I. Steinzen, L. Suhl, and N. Kliewer, “Branching strategies to improve regularity of crew schedules in ex-urban public transit,” *ATMOS 2007—7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, C. Liebchen, R.K. Ahuja, and J.A. Mesa, (Editors), Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.
- [26] I. Sungur, Y. Ren, F. Ordóñez, M. Dessouky, and H. Zhong, A model and algorithm for the courier delivery problem with uncertainty, *Transp Sci* 44 (2010), 193–205.
- [27] C. Tarantilis, F. Stavropoulou, and P.P. Repoussis, Template-based tabu search algorithm for the consistent vehicle routing problem, *Expert Syst Appl* 39 (2012), 4233–4239.
- [28] H. Zhong, R.W. Hall, and M. Dessouky, Territory planning and vehicle dispatching with driver learning, *Transp Sci* 41 (2007), 74–89.



A multi-criteria large neighbourhood search for the transportation of disabled people

Fabien Lehuédé^{1*}, Renaud Masson¹, Sophie N Parragh², Olivier Péton¹ and Fabien Tricoire^{2,3}
¹LUNAM Université, École des Mines de Nantes, Nantes, France; ²University of Vienna, Vienna, Austria; and ³NICTA/UNSW, Sydney, Australia

This paper addresses the problem of optimizing the transportation of disabled persons from home to specialized centres or schools. It is modelled as a Dial-a-ride problem (DARP), where several people share the same destination. Particular emphasis is placed on the objective function in order to consider several potentially conflicting interests. We propose a multi-criteria model from Multi-attribute Utility Theory based on the Choquet integral. The resulting multi-criteria DARP is then solved with a large neighbourhood search algorithm. This method includes classical destroy and repair heuristics as well as new operators exploiting the shared destination feature and criterion-specific operators. The algorithm is evaluated on a set of 14 real-world instances in the field of health care logistics, with up to 200 requests and 51 destination points.

Journal of the Operational Research Society advance online publication, 8 May 2013; doi:10.1057/jors.2013.17

Keywords: multi-criteria optimization; dial-a-ride problem; large neighbourhood search; Choquet integral; health care logistics

1. Introduction

Dial-a-ride problems (DARP) arise, for example, in patient transportation, in transportation on demand in rural areas or in public services for the disabled. A number of transportation requests have to be served by a given fleet of vehicles. Each request is associated with a pickup and a delivery location. In contrast to the vehicle routing problem or the pickup and delivery problem, DARP applications often concern the transportation of persons. They generally include additional constraints such as time windows or user ride time limits. Moreover, solutions to the DARP must be efficient from both financial and quality of service points of view.

Due to its application-oriented character, no standardized definition of the DARP exists. The objective functions usually encountered range from the maximization of the number of patients served to the minimization of user waiting times or routing costs. The constraints considered are usually tailored to the specific problem situation.

In this paper we consider the transportation of mentally disabled persons. These persons are able to get on and off vehicles without physical assistance, but due to their lack

of autonomy, they cannot use private means of transportation or the general public transportation system. For this reason, specialized transportation services with eight-seater minibuses have been put into place. We are interested in three main applications: (i) the transportation of adults to vocational rehabilitation centres, (ii) the transportation of disabled children to specialized schools, (iii) the transportation of disabled children to regular schools. The difference between cases (ii) and (iii) is that specialized schools may receive several dozens of children whereas regular schools generally receive only a few disabled children.

The organization of transport involves three types of actors: the *service consumers*, that is the persons transported who are particularly sensitive to the quality of service; the *service providers*, that is the transportation companies that are focused on the logistics aspects (minimizing the number of vehicles used, the distance travelled, etc); the *service coordinators*, that is the centres or schools or local authorities who establish the transportation plans and are responsible for the travellers. In our approach, the decision makers are the service coordinators. In practice, they can be either the directors of rehabilitation centres or principals of specialized schools, or local authorities responsible for transportation to schools. Their objectives can be the minimization of travelling costs or the maximization of quality of service for the passengers. In any case, the decision making process seeks a trade-off between interests of all stakeholders.

*Correspondence: Fabien Lehuédé, LUNAM Université, École des Mines de Nantes, IRCCyN UMR CNRS 6597, 4 rue Alfred Kastler, Nantes cedex 3, 44300, France.

As far as travelling costs are concerned, transportation companies generally have abstruse cost policies and variable practices from one area to another. Thus, the decision makers know the global price charged by every service provider, but are unable to decompose it by client or even by route. The financial impact of every routing decision cannot be measured, so cost cannot be considered as an optimization criterion. The quality of service for one customer depends on several factors such as the user ride time, time window and waiting time. For these reasons, designing an efficient transportation plan corresponds to solving a multi-objective optimization problem. In this paper, rather than producing a large set of non-dominated solutions, we propose to combine ideas from multi-criteria decision making (MCDM) with a metaheuristic optimization algorithm in order to produce one compromise solution based on several criteria. This is only possible in a context where decision maker preferences are known in advance.

We model the transportation problem as a DARP with many pickup points and a few delivery points and propose a mixed integer programme and a multi-criteria model based on tangible factors, such as, for example, distance, total ride time, waiting time, utilization of the vehicle capacity, etc. One original feature of the model is the consideration of user waiting time at destinations; it corresponds to the time lag between the user arrival at the school or centre and its actual opening time. Given its success for several vehicle routing problems (Ropke and Pisinger, 2006b, 2010), and the pickup and delivery problem with time windows in particular (Ropke and Pisinger, 2006a), we solve the problem with a large neighbourhood search (LNS) algorithm. The algorithm is evaluated on a collection of real-world instances arising from vocational rehabilitation centres and schools in the area of Nantes (France). Original aspects of our work include the exploitation of common (shared) delivery points and the integration of the multi-criteria model in LNS.

The field of DARP has received considerable attention in the literature (Cordeau and Laporte, 2007; Parragh *et al*, 2010b). Many of the methods published are designed to solve real-world applications, the minimization of operating costs (usually dependent on the total distance travelled) being the predominant objective (Toth and Vigo, 1997; Cordeau and Laporte, 2003; Parragh *et al*, 2010a; Parragh and Schmid, 2013), while quality of service is typically ensured using hard constraints, for example maximum ride time. Quality of service-related objectives have also been considered. However, these are typically used instead of the cost objective, or in combination with a cost objective using a weighted sum approach (Madsen *et al*, 1995; Diana and Dessouky, 2004; Melachrinoudis *et al*, 2007). Pareto multi-objective approaches for solving DARP have been the subject of recent research. Parragh

et al (2009) concurrently consider the total operating costs and the mean user ride time, a quality of service-related objective; Lacomme *et al* (2009) concurrently minimize the total ride time, the total trip duration and the total waiting time; Paquette (2010) concurrently considers total operating costs, excess user ride time and waiting times on board a vehicle. Chevrier *et al* (2010) compare three algorithms (NSGA-II, IBEA and SPEA2) on a demand-responsive transport problem with three minimization objectives, respectively, related to fleet size, travel duration and delays at delivery points. Atahran *et al* (2011) propose a genetic algorithm based on NSGA-II with three objectives: minimizing transportation cost, minimizing users dissatisfaction (modelled by a weighted sum of time windows violation) and minimizing the total quantity of CO₂ emitted.

To the best of our knowledge, multi-objective DARP have exclusively been solved using either the Pareto approach or a weighted sum until now. In this paper, we rather follow a multi-criteria aggregation approach to deal with several concurrent objectives. It allows the computation of a single solution by using extensive *a priori* preference information from the decision makers. The use of preference models from MCDM in optimization is receiving more and more attention (Coello Coello, 2000; Branke *et al*, 2008; Fernandez *et al*, 2009). In particular, several recent interactive methods integrate preference models (Branke *et al*, 2008, 2009). This allows to help the decision makers choose among the large set of efficient solutions, by reducing the search space using preference information.

The MCDM model used in this paper is a utility model based on the Choquet integral (Grabisch and Labreuche, 2010). It is particularly attractive as it is able to model complex preferences, taking account not only of relative importances, but also of interactions between criteria. So far, only a few papers propose to integrate a utility model based on the Choquet integral into optimization algorithms: Lehuédé *et al* (2006a, b) propose to integrate the whole preference model in constraint programming. Then, some exact algorithms have been developed to optimize the Choquet integral in shortest path (Galand *et al*, 2010; Fouchal *et al*, 2011a, b) or spanning tree (Galand *et al*, 2010) problems. However, these approaches make some restrictive hypotheses on the preference model. Finally, Dubus *et al* (2009) use the Choquet integral to find balanced solutions in configuration problems. To our knowledge, this model has not been used in combination with metaheuristics nor in the context of vehicle routing.

The paper is organized as follows. In Section 2, we define the multi-criteria dial-a-ride problem (MCDARP). We detail the construction of four preference models, each based upon a different preference profile. Section 3 describes the LNS approach used to solve the problem.

It details the destroy and repair operators used to modify the current solution in every iteration and introduces a scheduling procedure that checks time feasibility. Numerical experiments on a collection of real-world instances are then reported in Section 4.

2. The MCDARP

We first define some notations and the decision variables of the considered optimization problem. We then introduce the MCDM preference model and present how the objective functions can be expressed as metrics for the MCDM model. We then provide several preference settings that reflect different preference profiles. The section ends with a complete description of the mathematical model.

2.1. Notations and variable definitions

We consider a set of requests that correspond to trips between pickup and delivery locations. Let R be the set of requests, P the set of pickup locations and D the set of delivery locations. We also define sets O and O' to represent the vehicles' initial and final depots. The set of available vehicles is finite and denoted by K . Each route is performed by a vehicle $k \in K$, of capacity C , that starts from a depot $o(k) \in O$ and ends at a depot $o'(k) \in O'$. For ease of exposition, we define the sets $N = P \cup D$ of all pickup or delivery nodes and the set $V = P \cup D \cup O \cup O'$ of all vertices. For every request $i \in R$, the pickup and delivery points are denoted by $p(i) \in P$ and $d(i) \in D$, respectively. Without loss of generality, one location is either a pickup or a delivery location but not both; whenever needed, one physical location can be represented by two distinct vertices. With each request $i \in R$ is associated a number q_i of persons to be transported; requests cannot be split, that is each pickup node must be visited exactly once. Every trip from location $i \in V \setminus O'$ to location $j \in V \setminus O$ is associated with a transportation time t_{ij} . The duration of a vehicle's route is limited to T and the passengers of a request $i \in R$ may not spend more than \bar{L}_i time units on a vehicle. Furthermore, each location $i \in N$ is associated with a service time s_i , reflecting the time needed for loading and unloading operations, and a time window $[e_i, l_i]$. In the case of a delivery location $i \in D$, l_i corresponds to the official opening time of the centre, and e_i is the time at which appropriate personnel is available to receive the persons dropped off at this centre. In the case of a pickup location the time window is computed on the basis of the user's maximum allowed ride time \bar{L}_i and of his minimum ride time $t_{p(i),d(i)}$. If a vehicle arrives at $d(i)$ before e_i , it has to wait until appropriate personnel is available; vehicle waiting time within the time window is not allowed. However, the passengers dropped off have to wait until the official opening time of the centre at l_i . The MCDARP

then consists in finding an optimal set of routes serving all requests while respecting time windows, route duration, capacity and waiting time restrictions.

In order to formulate the MCDARP we define the binary decision variables x_{ij}^k taking value 1 if the arc (ij) is traversed by vehicle k and 0 otherwise. For ease of exposition, we also define variables y_i^k , taking value 1 if request i is serviced by vehicle k and 0 otherwise. Finally, we define the following four families of continuous variables:

- A_i^k the time of arrival of vehicle k at vertex $i \in V$
- B_i^k the beginning of service of vehicle k at location $i \in N$
- L_i the ride time of request $i \in R$
- Q_i^k the number of passengers of vehicle k when leaving location $i \in N$

2.2. The multi-criteria framework

The MCDM research field aims at providing models, software and processes to model the preferences of an expert or decision maker over a set of criteria. We consider here a model with a set of \aleph criteria, organized in a single level of aggregation. Preferences on solutions are modelled considering their values over a set of metrics, denoted Ω_i , $i \in \{1, \dots, \aleph\}$, which have a similar purpose as objective functions in our context. An essential step of the approach consists in matching those heterogeneous metrics on a common satisfaction scale: $[0, 1]$ in our case. According to Multi-Attribute Utility Theory (MAUT, Keeney and Raiffa, 1976), we introduce partial utility functions $u_i: \Omega_i \rightarrow [0, 1]$, which establish commensurateness of the criteria. The values z_i returned by these functions indicate the level of satisfaction of the decision maker for each criterion independently (also called *score*). Considering a solution x , characterized by a vector $(x_1, \dots, x_{\aleph}) \in \Omega = \Omega_1 \times \dots \times \Omega_{\aleph}$, the overall assessment of x is given by

$$U(x) = H(u_1(x_1), \dots, u_{\aleph}(x_{\aleph}))$$

where $H: [0, 1]^{\aleph} \rightarrow [0, 1]$ is the aggregation function. The preference relation \succsim between two solutions x and x' is such that $x \succsim x' \Leftrightarrow U(x) \geq U(x')$. We propose to use the two-additive version of the Choquet integral. This function models not only criteria importances, but also positive and negative interactions between criteria, which often appear in complex preferences.

The two-additive Choquet integral is defined for a vector $(z_1, \dots, z_{\aleph}) \in [0, 1]^{\aleph}$ (Grabisch, 1997):

$$H(z_1, \dots, z_{\aleph}) = \sum_{i=1, \dots, \aleph} \left(\phi_i - \frac{1}{2} \sum_{j \neq i} |I_{ij}| \right) z_i + \sum_{I_{ij} > 0} I_{ij} \min(z_i, z_j) + \sum_{I_{ij} < 0} |I_{ij}| \max(z_i, z_j) \quad (1)$$

where ϕ_i is the relative importance of criterion i and I_{ij} is the interaction between criteria i and j . The definition of $H(z_1, \dots, z_N)$ is composed of a linear part followed by a conjunctive and a disjunctive part. As the importance ϕ_i of a criterion includes both its importance alone and in conjunction with other criteria, half of the interactions in which i interferes are subtracted from ϕ_i in the linear coefficient of z_i . In the conjunctive part, a positive interaction index denotes *complementarity* between two criteria: the min operator expresses that for a high overall score, two criteria should both have a high score. Conversely, in the disjunctive part, two criteria with a negative interaction index produce a high score as soon as one criterion is well satisfied. In this case, there is *substitutability* between these criteria. Note that the weights of z_i , $\min(z_i, z_j)$ and $\max(z_i, z_j)$ in (1) are non-negative and they sum up to one.

As recommended in MCDM, the coefficients and functions of the model should not be asked to the decision maker but built following a preference elicitation process (mainly based on solution comparisons) and with the support of a software. Establishing a MAUT model based on a two-additive Choquet integral can be achieved using the MYRIAD software (Labreuche and Lehuédé, 2005) or the KAPPALAB library (Grabisch *et al.*, 2008). An extensive description of this preference model and of its use in preference modelling can be found in the survey of Grabisch and Labreuche (2010).

2.3. Metrics for the MCDARP

Many possible quantitative criteria exist for comparing solutions of DARP. We can distinguish cost-related criteria (based on the number of vehicles used, the total distance travelled, etc) and quality of service-related criteria (based on ride time or waiting time considerations) reviewed in Paquette *et al.* (2009).

We identify and consider five relevant objective functions for the MCDARP and use them to build the multi-criteria model. Let X be the set of all solutions to the MCDARP and $x \in X$. We define:

$$f_{mrt}(x) = \max_{i \in R} (L_i - t_{p(i),d(i)}) \quad (\text{maximum excess ride time}),$$

$$f_{art}(x) = \frac{1}{\sum_{i \in R} q_i} \sum_{i \in R} q_i (L_i - t_{p(i),d(i)}) \quad (\text{average excess ride time}),$$

$$f_{wtd}(x) = \sum_{i \in R} q_i (l_{d(i)} - B_{d(i)}^k) \quad (\text{total waiting time at destination}),$$

$$f_{tt}(x) = \sum_{k \in K} \sum_{ij \in V} x_{ij}^k t_{ij} \quad (\text{total travel time}),$$

$$f_v(x) = \sum_{k \in K} \sum_{i \in P} x_{o(k),i}^k \quad (\text{number of vehicles used}).$$

Objective functions f_{mrt} and f_{art} are based on the notion of *excess ride time*, which denotes the difference between the actual ride time of a request i and the time of a direct trip between its pickup $p(i)$ and its delivery $d(i)$. The maximum and average excess ride time over all requests are also used as objective functions by Spada *et al.* (2005). Note that since $\sum_{i \in R} t_{p(i),d(i)}$ is a constant value, minimizing the average excess ride time is equivalent to minimizing the sum $\sum_{i \in R} q_i L_i$ of all ride times. Objective function f_{wtd} measures the waiting time of users at destination (Paquette, 2010). Functions f_{tt} and f_v are used as indicators for the cost of a solution, which is likely to include a fixed part related to the number of vehicles used and a proportional part related to the total travelling time of all vehicles. Note that if a vehicle arrives at destination $d(i)$ before the opening e_i of the time window, the time of service $B_{d(i)}^k$ is e_i . Thus the waiting time $B_{d(i)}^k - A_{d(i)}^k$ in the vehicle is incorporated in the total travel time function t_{tt} whereas the waiting time at destination $l_{d(i)} - B_{d(i)}^k$ is considered in f_{wtd} .

The proposed model is defined such that it can be applied to a large variety of instances without modifications. Contrary to an objective function, a metric of the MCDM model should have a range of values as independent as possible of the instance size. Ideally, the satisfaction level provided by a given value for a metric should be the same for all instances. Hence, a normalization step is needed to build a metric upon an objective function.

Clearly, apart from the maximum and average excess ride times, the values taken by the objective functions depend on the size of the instance. Let $Q = \sum_{i \in R} q_i$ be the total number of passengers. In order to obtain instance independent metrics, we normalize f_{wtd} and f_{tt} by dividing them by Q .

Further analysis is required in order to produce a metric for the number of vehicles used. Under the hypothesis of a homogeneous fleet, it has been observed that the average occupancy rate over all used vehicles is a relevant metric to express preferences. It is equivalent to the number of used vehicles. The average occupancy rate over all vehicles used is defined by:

$$a_v(x) = \frac{Q}{f_v(x)C}, \quad \forall x \in X.$$

Note that the average occupancy rate can be more than one for the DARP, since one vehicle may load and unload more than C passengers during its trip.

We now introduce the following set of metrics, reflecting the five objective functions previously described:

$$(\Omega_1) \quad m_{mrt}(x) = f_{mrt}(x) \quad (\text{maximum excess ride time}),$$

$$(\Omega_2) \quad m_{art}(x) = f_{art}(x) \quad (\text{average excess ride time}),$$

$$(\Omega_3) \quad m_{wid}(x) = \frac{f_{wid}(x)}{Q} \quad (\text{average waiting time at destination}),$$

$$(\Omega_4) \quad m_{tt}(x) = \frac{f_{tt}(x)}{Q} \quad (\text{average travel time}),$$

$$(\Omega_5) \quad m_v(x) = \frac{Q}{f_v(x)C} \quad (\text{average occupancy rate of used vehicles}).$$

2.4. An MCDM model for the MCDARP

On the basis of the definitions of the metrics, a partial utility function has first been built for each criterion. We consider piecewise linear functions, that are used in the MACBETH approach (Bana e Costa *et al*, 2005), adapted to the Choquet integral (Grabisch and Labreuche, 2010).

Figure 1 presents the set of partial utility functions, one for each of the MCDARP criteria. For each criterion, a utility function provides a satisfaction level on scale [0, 1] for each possible value of its associated metric.

2.5. Objective functions based on decision maker preferences

In this paper, we consider four different aggregation models for the objective function. The first two models represent each a different profile of decision maker. These

models have been computed from two sets of solutions comparisons. The latter two are designed for control purposes and do not really reflect a specific decision maker.

The first model mimics the preferences of a *vehicle operator* for which the main focus is cost. This focus is translated into high weights ($\phi_{tt}=0.4$ and $\phi_v=0.45$) for criteria *average travel time* ($u_4(m_{tt}(x))$) and *average occupancy rate of used vehicles* ($u_5(m_v(x))$), respectively. Quality of service-oriented criteria (*maximum excess ride time*, *average excess ride time* and *average waiting time at destination*) have small individual criterion importance ($\phi_{mrt}=0.09$, $\phi_{art}=0.04$, $\phi_{wid}=0.02$). Moreover, all positive interactions involve at least one cost-related criterion. As a result, quality of service criteria are mostly expressed in conjunction with cost criteria. In other words, if a solution is good from the cost perspective, improving quality of service will increase the objective function value. If it is bad on cost criteria, improving quality of service will improve the objective function value to a much lesser extent. We denote the model for the *vehicle operator* by μ_{cost} in the following.

The second model mimics the preferences of a *user-oriented decision maker*. Conversely to the *vehicle operator*, the *user-oriented decision maker* puts all quality of service criteria before the cost criteria, translating into a high preference value for *maximum excess ride time* ($\phi_{mrt}=0.35$), followed by *average excess ride time* ($\phi_{art}=0.3$) and *waiting time at destination* ($\phi_{wid}=0.2$). The utility functions based

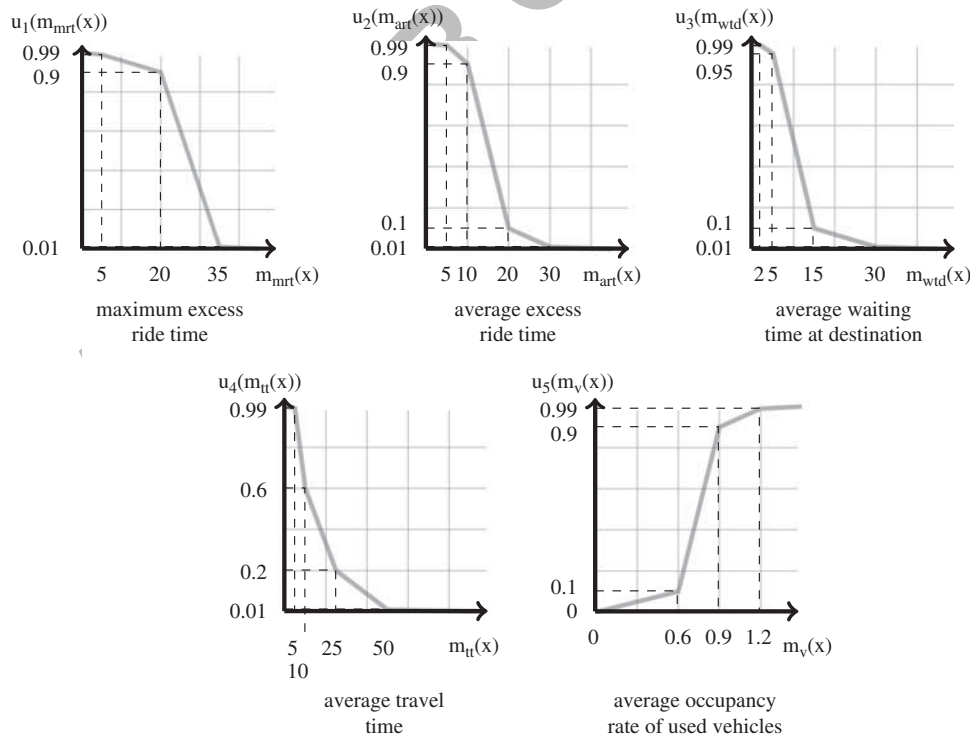


Figure 1 Partial utility functions for MCDARP criteria.

on cost-related criteria only receive a weight of $\phi_{tt} = \phi_v = 0.075$. In terms of interactions between criteria, the *user-oriented decision maker* attributes the highest weight to *maximum excess ride time* in combination with *average excess ride time* ($I_{mrt,art} = 0.3$). No impact on the overall utility is associated with interactions between *average excess ride time* or *waiting time at destination* and any of the cost-related criteria ($I_{art,tt} = I_{art,v} = I_{wtd,tt} = I_{wtd,v} = I_{tt,v} = 0$). In the following, we denote this model as μ_{QoS} .

Models three and four are established for control purposes. The third model, denoted μ_{avg} , corresponds to a pure arithmetic average of all criteria, without any interaction between criteria. The fourth model, denoted μ_{bal} , looks for balanced solutions, assumes some interactions between criteria and puts equal weights on all interactions.

For each model, the relative importance ϕ_i and the interactions I_{ij} in the definition of $H(z_1, \dots, z_8)$ are given in Tables 1 and 2, respectively.

2.6. A mathematical formulation of the MCDARP

Using the above-introduced notation, we are now able to formulate the MCDARP:

$$\max z = H(u_1(m_{mrt}(x)), u_2(m_{art}(x)), u_3(m_{wtd}(x)), u_4(m_{tt}(x)), u_5(m_v(x))) \quad (2)$$

subject to:

$$\sum_{k \in K} y_i^k = 1 \quad \forall i \in R, \quad (3)$$

$$\sum_{j \in N} x_{p(i),j}^k = y_i^k \quad \forall i \in R, k \in K, \quad (4)$$

$$\sum_{j \in N} x_{p(i),j}^k - \sum_{j \in N} x_{j,d(i)}^k = 0 \quad \forall i \in R, k \in K, \quad (5)$$

Table 1 Criteria importance

	ϕ_{mrt}	ϕ_{art}	ϕ_{wtd}	ϕ_{tt}	ϕ_v
μ_{cost}	0.09	0.04	0.02	0.4	0.45
μ_{QoS}	0.35	0.3	0.2	0.075	0.075
μ_{avg}	0.2	0.2	0.2	0.2	0.2
μ_{bal}	0.2	0.2	0.2	0.2	0.2

Table 2 Interactions between criteria

	$I_{mrt,art}$	$I_{mrt,wtd}$	$I_{mrt,tt}$	$I_{mrt,v}$	$I_{art,wtd}$	$I_{art,tt}$	$I_{art,v}$	$I_{wtd,tt}$	$I_{wtd,v}$	$I_{tt,v}$
μ_{cost}	0	0	0.08	0.1	0	0.02	0.05	0	0.04	0.5
μ_{QoS}	0.3	0.2	0.1	0.1	0.2	0	0	0	0	0
μ_{avg}	0	0	0	0	0	0	0	0	0	0
μ_{bal}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

$$\sum_{j \in P \cup \{o'(k)\}} x_{o(k),j}^k = 1 \quad \forall k \in K, \quad (6)$$

$$\sum_{i \in D \cup \{o(k)\}} x_{i,o'(k)}^k = 1 \quad \forall k \in K, \quad (7)$$

$$\sum_{i \in V} x_{ij}^k - \sum_{i \in V} x_{ji}^k = 0 \quad \forall j \in N, k \in K, \quad (8)$$

$$x_{ij}^k = 1 \Rightarrow Q_j^k \geq Q_i^k + q_j \quad \forall i \in V, j \in P, k \in K \quad (9)$$

$$x_{ij}^k = 1 \Rightarrow Q_j^k \geq Q_i^k - \sum_{i' \in R, d(i')=j} y_{i'}^k q_{i'} \quad (10)$$

$$\forall i \in N, j \in D, k \in K,$$

$$0 \leq Q_i^k \leq C \quad \forall i \in N, k \in K, \quad (11)$$

$$x_{ij}^k = 1 \Rightarrow A_j^k = B_i^k + s_i + t_{ij} \quad \forall i, j \in V, k \in K, \quad (12)$$

$$B_i^k \geq A_i^k \quad \forall i \in V, k \in K, \quad (13)$$

$$\sum_{i \in N} Q_i^k x_{ij}^k \geq 1 \wedge A_j^k \geq e_j \Rightarrow B_j^k = A_j^k \quad \forall j \in N, k \in K, \quad (14)$$

$$y_i^k (B_{d(i)}^k - B_{p(i)}^k - s_{p(i)}) \leq L_i \quad \forall i \in R, k \in K, \quad (15)$$

$$t_{p(i),d(i)} \leq L_i \leq \bar{L}_i \quad \forall i \in R, \quad (16)$$

$$B_{o'(k)}^k - B_{o(k)}^k \leq T \quad \forall k \in K, \quad (17)$$

$$e_i \leq B_i^k \leq l_i \quad \forall i \in V, k \in K, \quad (18)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall ij \in V, k \in K, \quad (19)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in R, k \in K, \quad (20)$$

$$Q_i^k \geq 0 \quad \forall i \in N, k \in K. \quad (21)$$

The objective function (2) minimizes a combination of the five above-introduced criteria. Constraints (3) guarantee that every request is served by one vehicle. Constraints (4) and (5) ensure that the pickup and delivery locations of a request are visited by the same vehicle. Constraints (6) and (7) make sure that every vehicle leaves its initial depot exactly once and returns to its end depot. If vehicle k is not used, constraints (6) and (7) are satisfied by an artificial route with cost 0 from $o(k)$ to $o'(k)$. Constraints (8) guarantee connectivity, that is every pickup and every delivery location is entered and left. Constraints (9) set the load variables. Constraints (10) ensure that the correct number of persons leave the vehicle at a delivery location. Constraints (11) guarantee that the vehicle capacity is not exceeded. Constraints (12) set the arrival time variables and constraints (13) ensure

that the beginning of service cannot start before the vehicle has arrived. Constraints (14) prevent a vehicle from waiting within the time window if it arrives at a pickup location i with passengers on board. Constraints (15) and (16) set the ride time of request i and ensure that the maximum ride time is satisfied. Constraints (17) ensure that the route duration limit is not exceeded. Constraints (18) guarantee time window compliance. Constraints (9), (10), (12), (14), (15) can be easily linearized. The interested reader may read Parragh (2009) for more details.

3. Solution approach

We propose an LNS algorithm for the MCDARP. LNS has been introduced by Shaw (1998) who uses it in a constraint programming framework in order to solve the vehicle routing problem with time windows (VRPTW). A similar principle is introduced by Schrimpf *et al* (2000) under the name *ruin and recreate*, and applied to the travelling salesman problem, the VRPTW and network optimization. Ropke and Pisinger (2006a) introduce adaptive large neighbourhood search (ALNS). In their version of the algorithm an adaptive mechanism is used to bias the selection among several destroy and repair operators in favour of those that performed well in previous iterations. They use it to solve the pickup and delivery problem with time windows. In Pisinger and Ropke (2007) a general-purpose LNS method for a wide range of routing problems, based on Ropke and Pisinger (2006a), is presented. These problems include the capacitated vehicle routing problem, the VRPTW, the site-dependent vehicle routing problem, the multi-depot vehicle routing problem and the open vehicle routing problem. A generalization of all these problems is considered, and each single problem is converted to the generalized model, then solved with the same ALNS method. Experimental results are provided for standard benchmark instances for each of the five considered problems, which show ALNS to outperform previous approaches.

All these contributions and more are reviewed in the recent survey by Ropke and Pisinger (2010). In the last years, LNS has been used extensively to solve transportation problems but the method has also been applied to planning and scheduling problems (see Cordeau *et al* (2010) for example).

An iteration of LNS consists in selecting a destroy–repair pair, applying it to the incumbent solution, and deciding to accept the newly generated solution as incumbent or not. In addition, the new incumbent is compared to the best solution found so far, and replaces it if relevant. These steps are repeated a number of times, until a given stopping criterion is met. Naturally, the design of appropriate destroy and repair heuristics is crucial for the global efficiency of the

method. Each pair of such operators yields a different large neighbourhood.

Our LNS is based on the ALNS of Ropke and Pisinger (2006a). Following their findings, we also use a simulated annealing scheme in the acceptance decision. However, we refrain from using an adaptive layer to guide the selection of the neighbourhood operators. We rather select them randomly, all neighbourhoods being equiprobable. Extensive tests showed that, in our case, no significant advantage could be identified when employing the adaptive mechanism. Algorithm 1 gives an overview of the LNS method developed for the MCDARP.

Algorithm 1 LNS

```

1 generate a solution  $s$ 
2  $s_{best} \leftarrow s$ 
3 repeat
4   randomly select a destroy-repair pair  $(d, \lambda r)$ 
5    $s' \leftarrow r(d(s))$ 
6   if  $s'$  is better than  $s_{best}$  then
7      $s_{best} \leftarrow s'$ 
8   end if
9   if  $s'$  meet the acceptance criteria then
10     $s \leftarrow s'$ 
11  end if
12 until some stopping criterion is met
13 return  $s_{best}$ 

```

In the following, we describe the different destroy and repair operators grouped according to *literature-based*, *problem-specific* and *criterion-oriented* operators. In each of the destroy operators, the number of requests to be removed from the solution, denoted as ξ , has to be determined. On the basis of preliminary tests, we choose ξ randomly in $(0.05|R^p|, 0.4|R^p|)$ following a uniform distribution, where R^p denotes the set of currently planned requests, usually $R^p = R$. In our case several requests share the same destination. When removing a request i from its route k , $p(i)$ is always removed but $d(i)$ is only removed if and only if there are no other requests in k going to the same destination. This is systematically tested in all destroy operators, with the notable exception of the *route removal heuristic* (see Section 3.2.3), that removes whole routes from the solution.

The different repair operators are either used in a deterministic or randomized version. A classical way of randomizing an operator is to apply *noise* in order to favour diversification. In our case noise is applied by multiplying evaluated objective function values by a noise factor. Let $H(j-1, i, j, k)$ denote the objective function value if currently non-routed request i is inserted into route k between $j-1$ and j . $H(j-1, i, j, k)$ is multiplied by a noise factor randomly determined in the interval $[l_{noise}, u_{noise}]$. On the

basis of preliminary experiments we set $l_{noise} = 0.99$ and $u_{noise} = 1.01$; the probability for using noise in the current LNS iteration is set to 0.5.

3.1. Operators from the literature

All operators described in this section correspond to operators proposed and employed by other authors (eg, Ropke and Pisinger, 2006a; Prescott-Gagnon *et al*, 2009) in the context of LNS or ALNS algorithms. They can be used in our LNS by simply using the Choquet integral instead of the usual objective function. However, given the complicated structure of our objective function, delta cost evaluations are unfortunately not possible.

3.1.1. Random removal. The random removal operator simply selects requests randomly. More precisely, all planned requests are sorted randomly, then the ξ first of them are removed from the solution.

3.1.2. Worst removal. This heuristic was introduced by Ropke and Pisinger (2006a). For each served request i in the current solution, we compute the objective value obtained when removing i . Requests leading to better partial solutions are associated with a higher selection probability. This process is repeated until ξ requests have been removed.

3.1.3. Distance-related removal. Related removal heuristics (Shaw, 1998) consist in removing a set of similar requests. We implement the version of Ropke and Pisinger (2006a): An initial request i is chosen at random and removed from the current solution. Then, a measure r_{ij} is computed to evaluate the *relatedness* of i with each request $j \neq i$. The next request to remove is chosen at random in the remaining requests, with a higher probability for requests with a low r_{ij} . Finally, the initial request for the next iteration is selected at random among the removed requests. This process is repeated until a predetermined number ξ of requests have been removed.

The *distance-related removal* operator evaluates the relatedness by comparing the location of pickup and delivery locations. For every pair (i, j) of requests, $r_{ij} = 1/4(t_{p(i),p(j)} + t_{d(i),p(j)} + t_{p(i),d(j)} + t_{d(i),d(j)})$.

3.1.4. Time-related removal. The time-related removal operator has been proposed by Prescott-Gagnon *et al* (2009) for the VRPTW. It removes customers that are serviced at about the same time. A specific time t_s is first randomly selected as a seed, then the customers

are ordered increasingly according to a value v_i defined as

$$v_i = \begin{cases} e'_i - t_s & \text{if } t_s \leq e'_i \\ 0 & \text{if } e'_i \leq t_s \leq l'_i \\ t_s - l'_i & \text{if } t_s > l'_i. \end{cases}$$

The interval $[e'_i, l'_i]$ corresponds to the time interval within which vertex i can be visited along its route in the current solution in a feasible way. We extended the time operator of Prescott-Gagnon *et al* (2009) to DARP, using $v'_i = \min(v_{p(i)}, v_{d(i)}) + \varepsilon \max(v_{p(i)}, v_{d(i)})$ as the distance measure to t_s (ε being a very small value). In this operator, ξ requests are removed at random with a higher probability for requests with a small v'_i .

3.1.5. Proximity-based removal. The proximity-based removal operator has also been proposed by Prescott-Gagnon *et al* (2009). It works similarly to the distance-related operator but considers both distance and time relatedness. For every pair (i, j) of requests, the according measure is defined as

$$r_{ij} = \text{dist}(ij) + \frac{1}{\text{time}(ij) + \text{time}(ji)},$$

where $\text{time}(ij) = \max(1, \min(l_{d(j)}, l_{d(i)} + t_{d(i),d(j)}) - \max(e_{d(j)}, e_{d(i)} + t_{d(i),d(j)}))$, $\text{dist}(ij) = 1/2T_{\max} \min(t_{p(i),p(j)}, t_{p(j),p(i)})$ and T_{\max} is a scaling factor set to the largest value of t_{ij} .

3.1.6. Best insertion. The best insertion heuristic is based on Ropke and Pisinger (2006a). It repeatedly inserts a request at the cheapest feasible position. Let $H(i, k)$ denote the objective value after inserting currently non-routed request i at its best possible position into route k ; and let $H(i, k) = 0$, if request i cannot be inserted into route k . Then, in every iteration, out of the set of currently unplanned requests R^u , the request i^* and the route k^* are chosen for which

$$(i^*, k^*) := \arg \max_{i \in R^u, k \in K} H(i, k). \quad (22)$$

Request i^* is then inserted into route k^* at its best position. This is repeated until all requests have been inserted or no more requests can be inserted. In our application, several requests generally share the same delivery location. Therefore, the delivery location $d(i^*)$ may already be part of the route k^* . In this case, the origin $p(i^*)$ is inserted before $d(i^*)$ and inserting this delivery location is not necessary.

3.1.7. Regret insertion. Following the findings of Ropke and Pisinger (2006a), we also use several regret insertion

heuristics. More precisely, we use a 2-regret, a 3-regret and a 4-regret insertion procedure. The regret for a request i is computed as follows:

$$H_{\text{regret}} = \sum_{r=2}^{\kappa} (H(i, 1) - H(i, r)), \quad (23)$$

and $\kappa \in \{2, 3, 4\}$, depending on the chosen version; $H(i, 1)$ denotes the objective function value if request i is inserted into its best route at the best position, while $H(i, 2)$ corresponds to the according value if request i was inserted into its second best route at the best position. In the case where a request cannot be inserted into a route r , the term is set to $H(i, r) = 0$. In every iteration the currently non-routed request associated with the largest regret value is inserted. If no feasible insertion position for a request i into a route r is found in the current iteration, insertion is not considered again in subsequent iterations. Otherwise, all objective function values have to be recomputed in every iteration.

3.2. Problem-specific operators

This section describes three destroy operators that exploit the structure of the MCDARP. The first two are based on the fact that some requests share the same destination, while the third operator aims at reducing the number of vehicles employed.

3.2.1. Delivery removal. This heuristic consists in iteratively removing from a route a destination node as well as all requests going to this destination. We select the destination with the lowest number of requests in the same route going to this destination.

3.2.2. Clustered pickup removal. First, a route is chosen at random. Then, within this route, a pickup location is randomly selected as the seed location s and its destination is determined. The cluster then consists of all vertices that (i) follow the seed location, (ii) are associated with the same destination and (iii) form an uninterrupted sequence. They are ordered according to their index and removed in a one-by-one fashion. Then, all pickup locations i that are currently on a different route are put into a list \mathcal{L} , ordered according to time closeness to the previous seed location s . Time closeness is computed by $|B_i - B_s|$. Then the pickup location at position $(\tau^{1.5} |\mathcal{L}|)$, where τ is a random number in $(0, 1]$ in the list \mathcal{L} becomes the next seed location, and the above is repeated until the number of requests removed meets the removal quantity ξ .

3.2.3. Route removal. The route removal iteratively selects a route at random and removes all its requests

from the solution, until ξ requests or more have been removed. One expected benefit is to remove a route with many different delivery points, so that all requests from this route can be reinserted into other routes already visiting the required delivery centre.

3.3. Criterion-oriented operators

All of the above operators rely on Choquet integral computations, so they are time consuming. In order to avoid these computations we propose several operators that are driven by only one out of the five criteria. Thus the criterion used can be considered as one parameter of the operator: we call these operators *parametrized* operators.

3.3.1. Parametrized worst removal. The parametrized worst removal operator acts similarly to the worst removal operator. However, the evaluation of each removed request is done on one specified criterion of the model and not on the objective function. Hence, one parametrized variant is defined for each criterion.

3.3.2. Parametrized best insertion. In addition to the standard best insertion heuristic described above, we also use several parametrized variants. In each variant, instead of the entire objective function value H , only the change in one of the different criteria is evaluated. All the different criteria have to be minimized. Therefore, in each variant, out of the set of currently unplanned requests R^u , the request i^* and the route k^* are chosen for which,

$$(i^*, k^*) := \arg \min_{i \in R^u, k \in K} \Delta(i, k), \quad (24)$$

where Δ corresponds to the change in either of the following, depending on the version: Δ_{mrt} , Δ_{art} , Δ_{wtd} , Δ_{tt} , Δ_v .

For three of these five criteria additional tie-breaking terms have been defined. In case of Δ_{wtd} , the total waiting time at a destination, and Δ_{mrt} , the maximum excess ride time, $\varepsilon = 0.01$ times the average ride time is added. In case of Δ_v , the delta costs for using an additional vehicle are set to $\max_{i \in N} (t_{ij}) |N|$, and the additional travel time needed to insert the according request is added for tie-breaking purposes.

3.3.3. Parametrized insertion with regret. Similarly to the parametrized best insertion heuristic, we also use a parametrized regret insertion algorithm. Here again, instead of the objective function H , one of the five criteria is used to compute the regret. The same tie-breaking rules as in the parametrized best insertion heuristic are employed. A major advantage of the parametrized version

lies in the small number of re-evaluations: only in the case of f_{mrt} do all delta insertion values have to be recomputed at every iteration. Otherwise, only those insertion values associated with a recently modified route have to be updated.

3.4. The scheduling algorithm

We use the route scheduling procedure outlined in Algorithm 2 in order to set the scheduling variables used by the objective function. At the same time, it is determined whether a given sequence corresponds to a feasible tour or not. It is based on the scheduling algorithm of Cordeau and Laporte (2003), which is itself based on the notion of forward time slack introduced by Savelsbergh (1992). We consider a particular ordered route (o, \dots, i, \dots, o') where o and o' denote the starting and terminating depot of the route, respectively. B_i denotes the beginning of service and F_i denotes the forward time slack at vertex i on the route.

Algorithm 2 Scheduling algorithm

```

1  set  $B_o := e_o$ 
2  initialize the forward time slack at the depot with
    $F_o := l_o - B_o$ 
3  for each vertex  $i$  on the route do
4    compute  $B_i := \max(B_{i-1} + s_{i-1} + t_{i-1,i}, e_i)$ 
5    if  $B_i > l_i$  then
6      return 'route infeasible' (time window check)
7    end if
8    increase total waiting time  $W := W + \max(0, e_i -$ 
    $(B_{i-1} + s_{i-1} + t_{i-1,i}))$ 
9    set  $F_o := \min(F_o, \max(0, l_i - B_i) + W)$ 
10 end for
11 set  $B_o := B_o + F_o$ 
12 update all  $B_i$ 
13 if  $B_o' - B_o > T$  then
14   return 'route infeasible' (duration check)
15 end if
16 for each zero load arc  $(ij)$  do
17   set  $W := \max(0, e_j - (B_i + s_i + t_{ij}))$ 
18   initialize the forward time slack at  $j$  with
    $F_j := W + l_j - B_j$ 
19   for each vertex  $v$  following  $j$  on the route do
20     set  $W := W + \max(0, e_v - (B_{v-1} + s_v + t_{v-1,v}))$ 
21     set  $F_j := \min(F_j, \max(0, W + l_v - B_v))$ 
22   end for
23   set  $F_j := \min(F_j, \max(0, W + B_o + T - B_j))$ 
24   set  $B_j := B_j + F_j$ 
25   update  $B_i$  at all vertices following  $j$ 
26 end for
27 compute all ride times  $L_i := B_{d(i)} - (B_{p(i)} + s_{p(i)})$ 
28 if some  $(L_i > \bar{L})$  then
29   return 'route infeasible' (ride time check)
30 end if

```

```

31 compute maximum excess ride time  $\max_i (L_i - t_{p(i),d(i)})$ 
32 compute total ride time  $\sum_i (q_i L_i)$ 
33 compute total waiting time at destination
    $\sum_i (q_i (l_{d(i)} - B_{d(i)}))$ 

```

In a first step, this algorithm plans all requests as early as possible (lines 1–10). Then, to reduce waiting times, the beginning of service at the initial depot is postponed as much as possible (lines 11–12). In contrast to Cordeau and Laporte (2003), we consider that the beginning of service at a pickup location cannot be postponed in order to reduce the ride time of the according request, unless the vehicle is empty. Hence, the third step of the algorithm (lines 16–26) acts similarly to steps one and two. It postpones the beginning of service at the first location after a zero load arc as much as possible in order to reduce ride times as well as waiting times at destinations.

4. Computational study

The heuristic is coded in C++ and executed on a Xeon CPU at 2.67 GHz with 24 GB of RAM. We produce three sets of destroy and repair operators, and test each of them in order to compare their respective usefulness. Therefore we have three LNS variants:

- *basic*: this variant uses all possible combinations of destroy operations among {random, worst, time-related, distance-related, proximity} with repair operations among {best insertion, 2-regret, 3-regret, 4-regret}. All those neighbourhoods exist in other (A)LNS methods for pickup and delivery problems.
- *advanced*: this variant contains all neighbourhoods from *basic* plus some MCDARP-specific ones exploiting the fact that some requests share the same delivery destination. More precisely, we add all possible combinations of destroy operations among {route, delivery, clustered pickups} with repair operations among {best insertion, 2-regret, 3-regret, 4-regret}.
- *parametrized*: this variant contains all neighbourhoods from *advanced* plus some criterion-specific neighbourhoods. More precisely, we add a parametrized *worst* destroy operator for each criterion, as well as parametrized best insertion, parametrized 2-, 3- and 4-regret repair operators for each criterion; then we combine each of these with all previously used operators, as well as with each other.

We now describe the data sets on which the experiments are run.

4.1. Instances

We evaluate our algorithm on a set of 14 real-world instances arising from distinct sources: specialized centres

for mentally disabled children, vocational rehabilitation centres for adults, and schools receiving both disabled and able-bodied children. In the first two sources, each delivery location is a common destination for dozens of requests. The instances considered correspond to a few centres wishing to pool the organization of their transports. The names of these instances start by ‘centres’ in the following of the paper.

In the case of transportation of disabled children to schools, most of these schools receive only a small number of handicapped children. In that case, the instances generally include all the schools of a given geographical area, possibly several dozens. The names of these instances start by ‘schools’.

The considered instances include between 55 and 200 requests. We only have to plan the trips from home to school or centre. Therefore, the pickup points are the personal addresses of the passengers and the delivery points are the centres or schools. All requests have distinct pickup points and correspond to individual demands. The number of delivery points is comprised between two and four for specialized schools or vocational rehabilitation centres and between 13 and 51 for all other schools. Every delivery point may have different opening hours. The time window $[e_i, l_i]$ for vehicle arrival is defined such that l_i corresponds to the beginning of the daily activity, that is school or specialized centre opening in the morning. The time window opening e_i is set to between 15 and 30 min prior to l_i , depending on the centre. The maximum ride time for passengers is set to 60 min, with one exception over all instances for a patient living far away from the centre.

We consider a service time of 2 min at pickup locations and of 5 min at delivery locations, regardless of the number of persons who get off the vehicle. Finally, the fleet of vehicles is considered to be homogeneous with a capacity of eight persons. We assume that the vehicles start and end at a dummy depot located at distance 0 from every pickup and every delivery location. The travelling time matrices between the different locations are asymmetric and satisfy the triangular inequality.

Instances are named ‘category- $|P|-|D|$ ’ such that the number of requests and delivery locations is reflected in their name. For example, instance centres-166-4 is of type ‘centres’ and consists of 166 requests and four delivery locations.

4.2. Comparison of the three LNS variants

The three variants use different neighbourhoods, each requiring a different CPU effort. For instance, the *basic* version systematically uses the Choquet integral, while the *parametrized* variant may also use faster single-criterion neighbourhoods. In addition, the *advanced* and *parametrized* versions use removal operators that remove several

requests at once instead of one by one. In order to produce total CPU efforts of the same order for each variant, we use different numbers of iterations: 10 000 for *basic*, 12 000 for *advanced* and 20 000 for *parametrized*. Those numbers were determined through preliminary experiments. For each multi-criteria model introduced in Section 2.4 we solve each instance 10 times with each variant. Tables 3–6 report these results, each table corresponding to a different multi-criteria model. In all tables, the best average objective values are bold-faced.

Evaluating the three different settings (*basic*, *advanced*, *parametrized*), we observe that in two out of four scenarios the *parametrized* variant obtains the best average results, that is for models μ_{cost} and μ_{QoS} . In the case of scenario μ_{avg} (see Table 5), the *basic* variant is better on average and

Table 3 Average values of H over 10 runs for aggregation scenario μ_{cost}

<i>Instance</i>	<i>Basic</i>	<i>Advanced</i>	<i>Parametrized</i>
centres-60-3	0.5981	0.5980	0.5971
centres-67-2	0.6849	0.6919	0.6955
centres-80-3	0.5810	0.5801	0.5900
centres-83-3	0.6359	0.6320	0.6353
centres-100-3	0.6315	0.6305	0.6425
centres-103-4	0.6222	0.6217	0.6225
centres-105-2	0.6809	0.6812	0.6773
centres-166-4	0.6250	0.6221	0.6283
schools-55-14	0.6095	0.6095	0.6095
schools-66-13	0.3607	0.3604	0.3609
schools-84-33	0.1415	0.1409	0.1407
schools-84-21	0.2051	0.2113	0.2126
schools-106-24	0.3218	0.3712	0.3784
schools-200-51	0.4045	0.4061	0.3990
Average	0.5073	0.5112	0.5136

Table 4 Average values of H over 10 runs for aggregation scenario μ_{QoS}

<i>Instance</i>	<i>Basic</i>	<i>Advanced</i>	<i>Parametrized</i>
centres-60-3	0.7167	0.7186	0.7201
centres-67-2	0.6998	0.7005	0.7094
centres-80-3	0.6466	0.6450	0.6430
centres-83-3	0.7951	0.7948	0.7950
centres-100-3	0.7352	0.7309	0.7300
centres-103-4	0.7426	0.7458	0.7468
centres-105-2	0.7989	0.7970	0.7976
centres-166-4	0.7060	0.6980	0.7113
schools-55-14	0.8713	0.8712	0.8711
schools-66-13	0.7875	0.7872	0.7871
schools-84-33	0.7658	0.7658	0.7661
schools-84-21	0.7896	0.7894	0.7894
schools-106-24	0.8017	0.8019	0.8022
schools-200-51	0.8006	0.8004	0.8009
Average	0.7612	0.7605	0.7621

Table 5 Average values of H over 10 runs for aggregation scenario μ_{avg}

Instance	Basic	Advanced	Parametrized
centres-60-3	0.6859	0.6853	0.6869
centres-67-2	0.7243	0.7237	0.7228
centres-80-3	0.6537	0.6566	0.6547
centres-83-3	0.7347	0.7342	0.7338
centres-100-3	0.7124	0.7103	0.7137
centres-103-4	0.7024	0.7030	0.7010
centres-105-2	0.7401	0.7410	0.7410
centres-166-4	0.6756	0.6732	0.6692
schools-55-14	0.8510	0.8510	0.8508
schools-66-13	0.6935	0.6935	0.6935
schools-84-33	0.6697	0.6696	0.6703
schools-84-21	0.6879	0.6874	0.6872
schools-106-24	0.7087	0.7084	0.7085
schools-200-51	0.7018	0.7011	0.7021
Average	0.7101	0.7099	0.7097

Table 6 Average values of H over 10 runs for aggregation scenario μ_{bal}

Instance	Basic	Advanced	Parametrized
centres-60-3	0.5342	0.5335	0.5369
centres-67-2	0.6115	0.6156	0.6102
centres-80-3	0.4947	0.4946	0.4902
centres-83-3	0.5405	0.5418	0.5437
centres-100-3	0.5662	0.5687	0.5663
centres-103-4	0.5623	0.5615	0.5603
centres-105-2	0.5854	0.5966	0.5869
centres-166-4	0.4916	0.4868	0.4888
schools-55-14	0.7423	0.7421	0.7422
schools-66-13	0.4584	0.4585	0.4583
schools-84-33	0.4277	0.4275	0.4272
schools-84-21	0.4482	0.4478	0.4479
schools-106-24	0.4822	0.4822	0.4825
schools-200-51	0.4695	0.4690	0.4713
Average	0.5296	0.5304	0.5295

in case of scenario μ_{bal} , the advanced variant obtains the best average results. If we have a closer look at the best method on a per instance level, the *parametrized* variant obtains the highest number (9) of best solutions per instance for scenario μ_{cost} ; for scenario μ_{QoS} it ties with the basic version with 7 per instance average best results. For scenario μ_{avg} it comes second with 6 per instance average best results. For scenario μ_{bal} , it shares the second place with the advanced variant (4 per instance average best results).

If we look at scenario μ_{cost} and instance centres-100-3, the *parametrized* variant is especially advantageous in comparison with the *basic* and the *advanced* variant. Attempting an interpretation of this result, we use pie charts to illustrate the components of the Choquet integral

function. Figure 2 represents the Choquet integral values for three solutions, x_a , x_b and x_c , of instance centres-100-3 (the metrics and utility values for these solutions are given in Table 7).

Each pie chart represents a different solution. Each slice represents a different term of the two-additive Choquet integral for that solution. Criteria individual importances are labelled v (average occupancy rate of used vehicles), tt (total travel time), art (average excess ride time), mrt (maximum excess ride time) and wtd (maximum waiting time at destination). Positive interactions are labelled (c_1, c_2) where c_1 and c_2 correspond to the interacting criteria, that is $c_1, c_2 \in \{v, tt, art, mrt, wtd\}$. Note that there are no negative interactions in any of our four preference models. The angle of each slice is proportional to the corresponding coefficient in Equation (1), while the ratio of shaded area in a slice is given by the z_i value associated with this slice (or the $\min(z_i, z_j)$ value for slices associated with positive interactions). The ratio of shaded area over the whole pie is the value of the Choquet integral for this solution.

Since we analyse solutions for model μ_{cost} , the criteria tt and v (total travel time and vehicle occupancy) and their interaction contribute the most to the overall Choquet integral value. The best out of these three solutions is x_c ; it is also the solution with the highest scores for cost-related terms such as v , tt and the interaction (tt, v) . But it is also the weakest solution for less important terms such as art , (art, v) , (mrt, v) , (mrt, tt) or (art, tt) . In other words, this solution sacrifices more quality of service than the others to achieve cost-related optimization. This produces a better Choquet integral value.

Analysing the results of the 10 random runs for each of the LNS versions on instance centres-100-3, an explanation of the performance differences can be attempted: the *basic* variant converges in five out of 10 cases to a solution that is associated with a very similar chart as solution x_c in Figure 2. The *advanced* variant converges to such a solution only in three out of 10 cases. In seven out of 10 runs solutions with a similar graphical representations as solutions x_a and x_b are obtained. Finally, in case of the *parametrized* variant, only three solutions are associated with lower Choquet integral values and charts similar to x_a and x_b while seven out of 10 runs attain a solution that is close or equal to solution x_c . This indicates that the parametrized variant produced, more often than the other two variants, solutions that optimize specific criteria at the expense of other secondary criteria.

This analysis strengthens the idea that substituting the time-consuming Choquet integral with single-criterion neighbourhood operations, as in the *parametrized* variant, can be productive if done properly. This is especially true in a context where sacrificing some criteria in order to improve other criteria leads to an overall improvement on the Choquet integral value. This is the case with scenarios

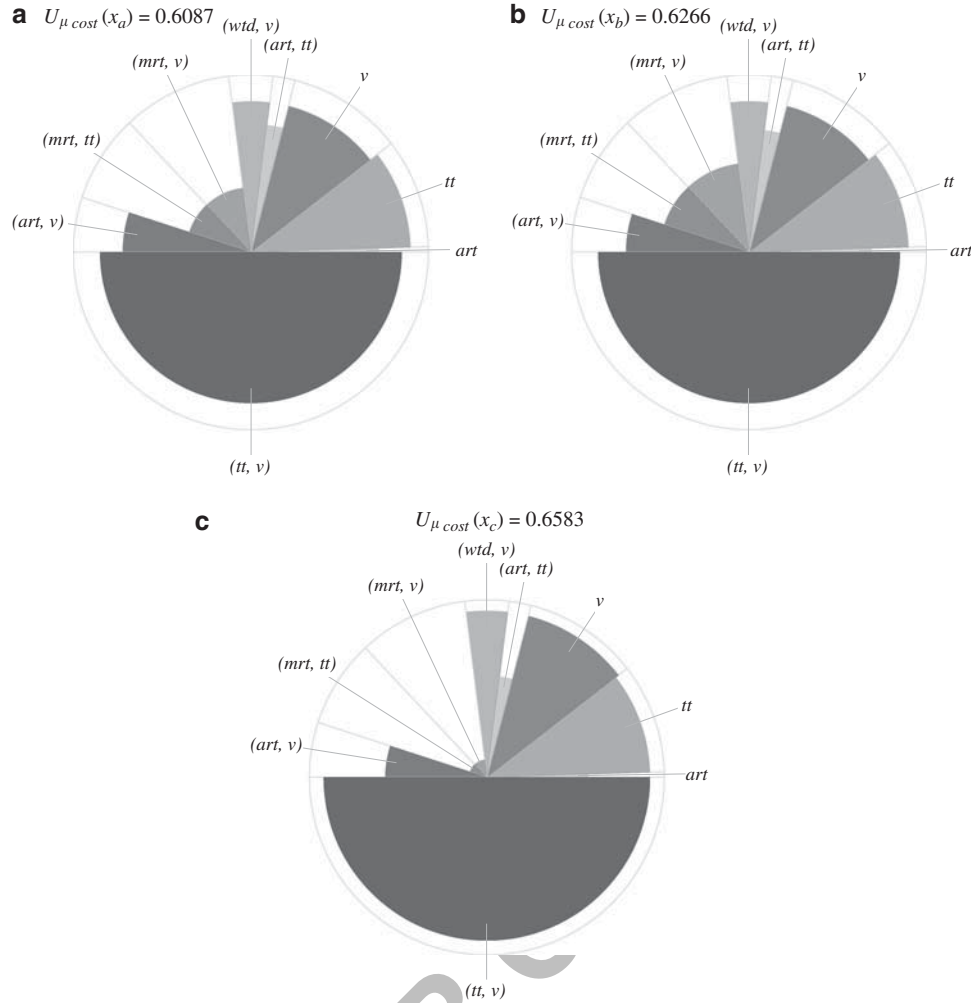


Figure 2 Representation of the Choquet integral value for different centres-100-3 solutions.

Table 7 Metrics and utility values for centres-100-3 solutions

	<i>mrt</i>	<i>art</i>	<i>wtd</i>	<i>tt</i>	<i>v</i>	$U_{\mu_{cost}}$
x_a						
<i>u</i>	0.133	0.52	0.823	0.809	0.722	0.6087
<i>m</i>	32.93	14.74	6.5	7.32	0.83	
x_b						
<i>u</i>	0.251	0.48	0.743	0.807	0.722	0.6266
<i>m</i>	30.93	15.25	7.44	7.35	0.83	
x_c						
<i>u</i>	0.01	0.327	0.912	0.845	0.881	0.6583
<i>m</i>	42.41	17.16	5.44	6.86	0.89	

μ_{cost} and μ_{QoS} , which explains why the parametrized variant works better than the other variants in these two scenarios. However, it is not the case with scenarios μ_{avg} and μ_{bal} , which also explains why the parametrized variant does not stand out when using these two scenarios.

However, it never provides bad results, making it a good default choice overall.

We also report the CPU times for the runs illustrated in the tables above. They are given in Tables 8–11. Again, per instances best average values are printed in bold. The tables show that the per instance CPU times of the different procedures all lie within the same order of magnitude and thus allow a fair comparison between the different variants.

The obtained CPU times also indicate that the consideration of common deliveries has an impact on the performance of the algorithm: apart from the centres-83-3 instance, centres instances with only a few delivery points are solved significantly faster than the schools instances for an equivalent number of requests. Nonetheless, the computation times observed are still acceptable for all instances. For the largest instances with 200 and 166 requests and run times of almost 2 h, the practical limitation of the method is reached.

Table 8 Average CPU time (s) over 10 runs for aggregation scenario μ_{cost}

Instance	Basic	Advanced	Parametrized
centres-60-3	39.4	36.1	45.6
centres-67-2	37.5	35.5	44.8
centres-80-3	74.8	68.1	80.6
centres-83-3	104.3	111.6	132.9
centres-100-3	154.7	136.4	144.2
centres-103-4	215.0	186.6	188.9
centres-105-2	173.3	177.6	186.5
centres-166-4	1347.9	1056.4	905.3
schools-55-14	34.3	54.8	45.4
schools-66-13	78.4	109.3	105.5
schools-84-33	195.8	169.1	202.3
schools-84-21	173.8	155.6	219.7
schools-106-24	520.7	385.7	367.7
schools-200-51	4691.5	3608.8	2915.5
Average	560.1	449.4	398.9

Table 10 Average CPU time (s) over 10 runs for aggregation scenario μ_{avg}

Instance	Basic	Advanced	Parametrized
centres-60-3	48.4	47.5	59.2
centres-67-2	49.9	52.3	59.6
centres-80-3	90.3	96.2	109.7
centres-83-3	178.7	180.1	194.3
centres-100-3	206.2	186.6	198.5
centres-103-4	281.6	235.5	272.2
centres-105-2	375.8	345.3	306.0
centres-166-4	1788.2	1969.4	1556.5
schools-55-14	36.1	53.5	51.4
schools-66-13	93.2	120.6	113.7
schools-84-33	182.9	199.7	201.4
schools-84-21	199.0	208.1	228.1
schools-106-24	680.3	530.3	481.7
schools-200-51	6967.3	6359.7	5216.5
Average	798.4	756.0	646.3

Table 9 Average CPU time (s) over 10 runs for aggregation scenario μ_{QoS}

Instance	Basic	Advanced	Parametrized
centres-60-3	53.7	51.1	59.7
centres-67-2	49.0	55.0	70.5
centres-80-3	106.5	87.8	90.5
centres-83-3	168.1	171.8	209.6
centres-100-3	198.7	209.1	235.9
centres-103-4	273.2	264.6	261.1
centres-105-2	288.4	321.9	282.5
centres-166-4	1919.3	1799.0	1734.5
schools-55-14	36.9	54.9	55.2
schools-66-13	88.7	120.2	107.2
schools-84-33	194.2	193.0	190.2
schools-84-21	217.0	215.5	221.1
schools-106-24	618.2	537.1	495.8
schools-200-51	6998.9	5711.8	5467.8
Average	800.8	699.5	677.2

Table 11 Average CPU time (s) over 10 runs for aggregation scenario μ_{bal}

Instance	Basic	Advanced	Parametrized
centres-60-3	51.9	44.4	52.9
centres-67-2	46.6	56.3	62.8
centres-80-3	81.7	89.5	93.5
centres-83-3	172.3	167.7	189.3
centres-100-3	205.8	186.0	187.7
centres-103-4	272.9	222.7	227.1
centres-105-2	299.3	250.1	255.3
centres-166-4	1737.0	1613.0	1642.4
schools-55-14	38.6	55.8	54.5
schools-66-13	90.3	93.4	110.9
schools-84-33	179.9	185.2	191.7
schools-84-21	194.4	197.8	217.3
schools-106-24	632.1	542.2	504.2
schools-200-51	6126.8	6165.1	5153.3
Average	723.5	704.9	638.8

5. Additional insight on the Choquet integral

We now analyse in further detail the obtained results, illustrating the trade-off between cost and quality of service and interpreting the Choquet integral values. For this purpose, we use one instance of each type: centres-83-3 and schools-106-24.

Figure 3 and Table 12 present a graphical interpretation of the Choquet integral values for two centres-83-3 solutions and the underlying per criterion utilities, respectively.

The left-hand pie chart (Figure 3(a)) depicts a good solution on instance centres-83-3 for the *vehicle operator* decision maker (model μ_{cost}). Cost-related criteria are more important than quality of service criteria. This is reflected

by the portions of the pie chart (Figure 3(a)) that are attributed to the *total travel time* criterion (tt) and the *average occupancy rate of used vehicles* criterion (v) as well as their interaction. The interaction term of these two criteria is associated with the largest area, indicating that *vehicle operator* decision makers prefer solutions that are good with respect to both criteria simultaneously. Both cost-related metrics have high values, while those related to quality of service (mrt, art, wtd) are low to medium. As a result the main component of the integral is high, and so are individual importances of tt and v . The individual importance of quality-of-service-related criteria and the positive interactions in which they are involved yield low values, but they have a small weight in the global objective anyway.

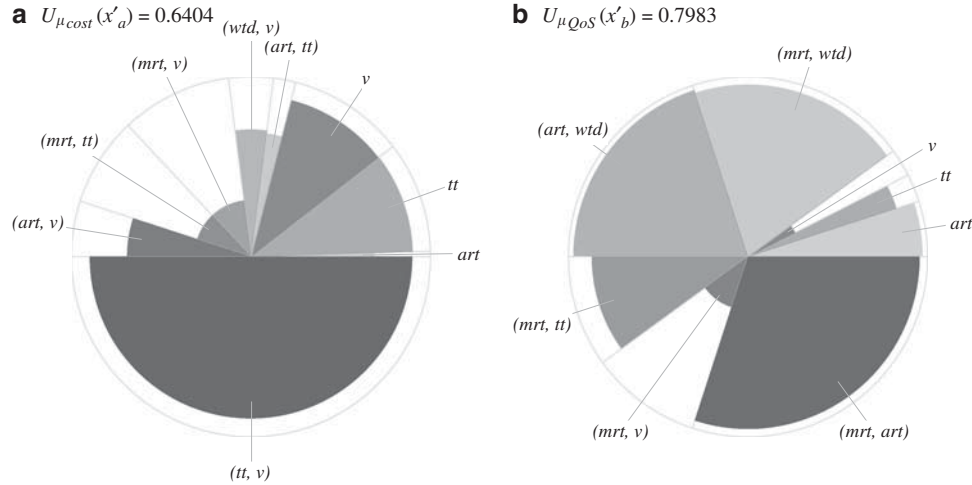


Figure 3 Graphical representation of the Choquet integral value for two centres-83-3 solutions.

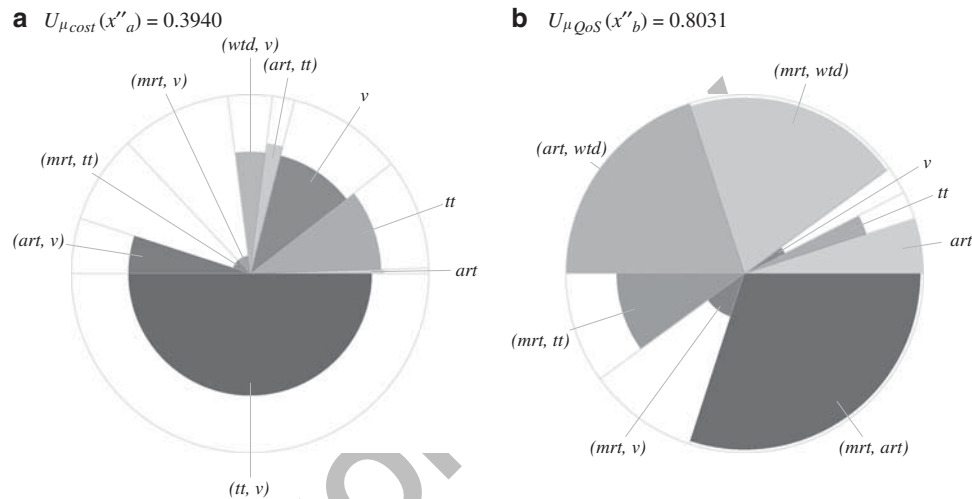


Figure 4 Graphical representation of the Choquet integral value for two schools-106-24 solutions.

Table 12 Per criterion metrics and utility values for two centres-83-3 solutions

	<i>mrt</i>	<i>art</i>	<i>wtd</i>	<i>tt</i>	<i>v</i>	<i>U</i>
x'_a						
<i>u</i>	0.099	0.4757	0.499	0.8121	0.8056	$U_{\mu_{cost}}(x'_a) = 0.6404$
<i>m</i>	33.5	15.30	10.31	7.28	0.86	
x'_b						
<i>u</i>	0.9138	0.9445	0.9764	0.7559	0.0864	$U_{\mu_{QoS}}(x'_b) = 0.7983$
<i>m</i>	17.7	7.53	3.02	8.0	0.52	

The right-hand pie chart (Figure 3(b)) depicts a good solution on instance centres-83-3 for the *user-oriented decision maker* (model μ_{QoS}). As defined above, interactions between quality of service criteria amount for the largest coefficients of the Choquet integral, thus for the

slices with the widest angles in the pie chart. Again, the interaction between *v* and *mrt* only has a very low filling rate. However, in this context the limiting factor is not *mrt* but *v*: a good solution with respect to quality of service comes at the price of a low utility value for the *average occupancy rate of used vehicles* ($u_5(m_v(x'_b)) = 0.0631$, as given in Table 12). Table 12 allows a further observation: while an emphasis of quality of service-related criteria comes at the price of a low utility for *average occupancy rate of used vehicles* (*v*), it only has comparably little impact on the utility for the *average travel time* (*tt*), the second cost-related criterion. This is a hint that quality of service-related criteria also benefit from good routing solutions, translating into shorter ride times as well as total travel times.

Figure 4 illustrates the Choquet integral values obtained for good solutions, for instance schools-106-24 in terms of,

Table 13 Per criterion metrics and utility values for two schools-106-24 solutions

	<i>mrt</i>	<i>art</i>	<i>wtd</i>	<i>tt</i>	<i>v</i>	<i>U</i>
x_a''						
<i>u</i>	0.01	0.5817	0.9584	0.5675	0.463	$U_{\mu_{cost}}(x_a'') = 0.394$
<i>m</i>	40.867	13.979	4.37	10.417	0.736	
x_b''						
<i>u</i>	0.9656	0.9944	0.9935	0.5122	0.0631	$U_{\mu_{QoS}}(x_b'') = 0.8031$
<i>m</i>	9.067	2.8	1.3	13.293	0.379	

respectively, cost (μ_{cost}) and quality of service (μ_{QoS}). The criterion values can be found in Table 13. In contrast to instances of type *centres*, the instances of type *schools* contain comparably more delivery points. This aspect translates into Choquet integral values for the *vehicle operator* decision maker that are well below those of the *centres* instances. Indeed, there is less room for improvement with respect to the cost aspect: more locations have to be visited along the routes while satisfying the same quality of service constraints. As for instances of type *centres*, cost minimization and therefore a better *average occupancy rate of used vehicles* (*v*) comes at the price of the utility for the *maximum excess ride time* (*mrt*) and vice versa.

6. Conclusions

Designing transportation plans for disabled people is a very complex task since several potentially conflicting interests have to be taken into consideration. An important preliminary step to optimization is then to model the decision maker's preferences. In this paper we model the transportation problem as an MCDARP and we propose a preference model able to accommodate various decision maker profiles.

The MCDARP is then solved with an LNS algorithm. We compare three configurations in terms of destroy and repair operators. The *basic* scheme includes operators that have been previously employed in the literature. The *advanced* scheme also includes additional operators exploiting the structure of the MCDARP. Finally, the *parametrized* configuration resorts to criterion-oriented operators, in addition to the ones from *basic* and *advanced*. The three configurations have been tested on a set of 14 real-world instances with up to 200 requests and 51 delivery points. Each variant provides the best average results for some instances and there is no clear winner, although the *basic* and *parametrized* yield more best average results than the *advanced* one. Still, no variant is dominated by the others.

From a preference modelling point of view, future research should involve the definition of preference models tailored to the type of instance solved and to the decision

maker evaluating the resulting solution. An interesting step toward a more generic decision support system would also be to embed the models and algorithms in an interactive method based for example on UTA^{GMS} or GRIP as proposed by Figueira *et al* (2008) and Branke *et al* (2009). These approaches combine the elicitation of the preference model and the search for a preferred solution. They have been extended to the Choquet integral by Angilella *et al* (2010). However, how partial utility functions can be integrated is still an open research question.

From a practical point of view, future research should also address the issue of very large instances. Local authorities are often responsible for several dozens to hundreds of schools. In practice, the problem is partitioned into several independent sub-areas, which are treated separately. A perspective is to perform a balanced separation to obtain improved solutions.

Finally, future research should also involve the integration of additional real-world requirements and objectives. Possible requirements involve the consideration of a heterogeneous fleet, different starting and ending depots, incompatibilities between passengers (passengers that should not be aboard the same vehicle at the same time), and inbound and outbound trips in the morning and in the afternoon; possible additional objectives involve regularity in terms of service times or driver consistency (Feillet *et al*, 2010).

Acknowledgements—We thank the Austrian Research Promotion Agency (FFG) under grants #826151 (program IV2Splus) and #822739 for sponsoring this work. The funding from the Austrian agency for international mobility and cooperation in education, science and research (ÖAD, Amadée project #FR 03/2009) is also greatly appreciated. This research was also supported by the Egidie Amadeus PHC Programme (project 19437ZF).

References

- Angilella S, Greco S and Matarazzo B (2010). Non-additive robust ordinal regression: A multiple criteria decision model based on the Choquet integral. *European Journal of Operational Research* **201**(1): 277–288.
- Atahran A, Lenté C and T'Kindt V (2011). Solution of a multicriteria dial-a-ride problem with an ecological measure and heterogeneous vehicles. In: *IESM 2011, International Conference On Industrial Engineering and Systems Management*, Metz, France, pp 1233–1242.
- Bana e Costa C, de Corte J and Vansnick J (2005). On the mathematical foundation of MACBETH. In: Figueira J, Greco S and Ehrgott M (eds). *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research and Management Science*, Springer: New York, pp 409–437.
- Branke J, Deb K, Miettinen K and Slowinski R (eds). (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches. Lecture Notes in Computer Science* 5252, Springer, Berlin.

- Branke J, Greco S, Slowinski R and Zelniewicz P (2009). Interactive evolutionary multiobjective optimization using robust ordinal regression. *Lecture Notes in Computer Science* **5467**: 554–568.
- Chevrier R, Liefvooghe A, Jourdan L and Dhaenens C (2010). On optimizing a demand responsive transport with evolutionary multi-objective approach. In *13th International IEEE Annual Conference On Intelligent Transportation Systems*, Madeira island, Portugal, 19–22 September, pp 575–580. IEEE.
- Coello Coello C (2000). Handling preferences in evolutionary multiobjective optimization: A survey. In *2000 Congress on Evolutionary Computation*, pp 30–37.
- Cordeau J-F and Laporte G (2003). A tabu search heuristic for the static multi-vehicle dial a-ride problem. *Transportation Research part B—Methodological* **37**(6): 579–594.
- Cordeau J-F and Laporte G (2007). The dial-a-ride problem: Models and algorithms. *Annals of Operations Research* **153**(1): 29–46.
- Cordeau J-F, Laporte G, Pasin F and Ropke S (2010). Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling* **13**(4): 393–409.
- Diana M and Dessouky MM (2004). A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B—Methodological* **38**(6): 539–557.
- Dubus J-P, Gonzales C and Perny P (2009). Choquet optimization using GAI networks for multiagent/multicriteria decision-making. *Algorithmic Decision Theory. Lectures Notes in Artificial Intelligence* **5783**: 377–389.
- Feillet D, Garaix T, Lehuédé F, Péton O and Quadri D (2010). A new consistent vehicle routing problem for the transportation of handicapped persons. Working Paper EMSE CMP-SFL 2010/9.
- Fernandez E, Lopez E, Bernal S, Coello Coello C and Navarro J (2009). Evolutionary multiobjective optimization using an out-ranking-based dominance generalization. *Computers & Operations Research* **37**(2): 390–395.
- Figueira J, Greco S, Mousseau V and Slowinski R (2008). Interactive multiobjective optimization using a set of additive value functions. *Lecture Notes in Computer Science* **5252**: 97–119.
- Fouchal H, Gandibleux X and Lehuédé F (2011a). A lower bound of the Choquet integral integrated within Martins' algorithm. In: Shi Y, Wang S, Kou G and Wallenius J (eds). *New State of MCDM in the 21st Century*. Volume 648 of *Lecture Notes in Economics and Mathematical Systems*. Springer: Berlin Heidelberg, pp 79–89.
- Fouchal H, Gandibleux X and Lehuédé F (2011b). Preferred solutions computed with a label setting algorithm based on Choquet integral for multi-objective shortest paths. In *SSCI 2011 MCDM—2011 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making*, Paris, France, April.
- Galand L, Perny P and Spanjaard O (2010). Choquet-based optimisation in multiobjective shortest path and spanning tree problems. *European Journal of Operational Research* **204**(2): 303–315.
- Grabisch M (1997). Alternative representations of discrete fuzzy measures for decision making. *International Journal of Uncertainty, Fuzziness, and Knowledge Based Systems* **5**(5): 587–607.
- Grabisch M and Labreuche C (2010). A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research* **175**(1): 247–286.
- Grabisch M, Kojadinovic I and Meyer P (2008). A review of methods for capacity identification in Choquet integral based multi-attribute utility theory: Applications of the Kappalab R package. *European Journal of Operational Research* **186**(2): 766–785.
- Keeney R and Raiffa H (1976). *Decision with Multiple Objectives*. Wiley: New York.
- Labreuche C and Lehuédé F (2005). MYRIAD: A tool suite for MCDA. In *EUSFLAT'05*, Barcelona, Spain, pp 204–209.
- Lacomme P, Quilliot A and Zhao X (2009). Dial-a-ride: A tentative resolution using a multiobjective resolution scheme based on ELS. In *CIE 2009. International Conference on Computers & Industrial Engineering*, IEEE, pp 1100–1105.
- Lehuédé F, Grabisch M, Labreuche C and Savéant P (2006a). Integration and propagation of a multi-criteria decision making model in constraint programming. *Journal of Heuristics* **12**(4–5): 329–346.
- Lehuédé F, Grabisch M, Labreuche C and Savéant P (2006b). MCS—A new algorithm for multicriteria optimisation in constraint programming. *Annals of Operations Research* **147**(1): 143–174.
- Madsen OB, Ravn HF and Rygaard JM (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research* **60**(1): 193–208.
- Melachrinoudis E, Ilhan AB and Min H (2007). A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research* **34**(3): 742–759.
- Paquette J (2010). *Mesure de la qualité de service et conception de tournées en transport adapté*. PhD thesis, HEC, Montréal.
- Paquette J, Cordeau J-F and Laporte G (2009). Quality of service in dial-a-ride operations. *Computers & Industrial Engineering* **56**(4): 1721–1734.
- Parragh S (2009). *Ambulance routing problems with rich constraints and multiple objectives*. PhD thesis, University of Vienna, Austria.
- Parragh S and Schmid V (2013). Hybrid column generation and large neighborhood search for the dial-a-ride problem. *Computers & Operations Research* **40**(1): 490–497.
- Parragh SN, Doerner KF, Hartl RF and Gandibleux X (2009). A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. *Networks* **54**(4): 227–242.
- Parragh SN, Doerner KF and Hartl RF (2010a). Variable neighborhood search for the dial-a-ride problem. *Computers & Operations Research* **37**(6): 1129–1138.
- Parragh SN, Doerner KF and Hartl RF (2010b). Demand responsive transportation. In: Cochran JJ (ed). *Wiley Encyclopedia of Operations Research and the Management Sciences*. Singapore: Wiley.
- Pisinger D and Ropke S (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research* **34**(8): 2403–2435.
- Prescott-Gagnon E, Desaulniers G and Rousseau L-M (2009). A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks* **54**(4): 190–204.
- Ropke S and Pisinger D (2006a). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* **40**(4): 455–472.
- Ropke S and Pisinger D (2006b). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research* **171**(3): 750–775.
- Ropke S and Pisinger D (2010). Large neighborhood search. In: Gendreau M and Potvin J-Y (eds). *Handbook of Metaheuristics*, 2nd edn, Springer: New York, Dordrecht, Heidelberg, London, pp 399–419.
- Savelsbergh MWP (1992). The vehicle routing problem with time windows: Minimizing route duration. *INFORMS Journal on Computing* **4**(2): 146–154.

Schrimpf G, Schneider J, Stamm-Wilbrandt H and Dueck G (2000). Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics* **159**(2): 139–171.

Shaw P (1998). Using constraint programming and local search methods to solve vehicle routing problems. In: Maher, M and Puget, J-F (eds), *Principles and Practice of Constraint Programming—CP98. Lecture Notes in Computer Science* 1520, Springer: Berlin Heidelberg, pp 417–431.

Spada M, Bierlaire M and Lieblich T (2005). Decision-aiding methodology for the school bus routing and scheduling problem. *Transportation Science* **39**(4): 477–490.

Toth P and Vigo D (1997). Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science* **31**(1): 60–71.

*Received July 2012;
accepted January 2013 after one revision*

AUTHOR COPY



A branch-and-cut-and-price approach for the pickup and delivery problem with shuttle routes



Renaud Masson^{a,c,*}, Stefan Ropke^b, Fabien Lehuédé^a, Olivier Péton^a

^a LUNAM Université, École des Mines de Nantes, IRCCyN, 4 rue Alfred Kastler, 44300 Nantes, France

^b Department of Transport, Technical University of Denmark, Bygningstorvet 116 Vest, DK-2800 Kgs. Lyngby, Denmark

^c CIRRELT, Département de Mathématiques et de Génie Industriel, Ecole Polytechnique de Montréal, Montréal H3C 3A7, Canada

ARTICLE INFO

Article history:

Available online 6 September 2013

Keywords:

Pickup and delivery problem
Transfers
Column generation
Branch-and-cut-and-price

ABSTRACT

The Pickup and Delivery Problem with Shuttle routes (PDPS) is a special case of the Pickup and Delivery Problem with Time Windows (PDPTW) where the trips between the pickup points and the delivery points can be decomposed into two legs. The first leg visits only pickup points and ends at some delivery point. The second leg is a direct trip – called a shuttle – between two delivery points. This optimization problem has practical applications in the transportation of people between a large set of pickup points and a restricted set of delivery points.

This paper proposes three mathematical models for the PDPS and a branch-and-cut-and-price algorithm to solve it. The pricing sub-problem, an Elementary Shortest Path Problem with Resource Constraints (ESPPRC), is solved with a labeling algorithm enhanced with efficient dominance rules. Three families of valid inequalities are used to strengthen the quality of linear relaxations. The method is evaluated on generated and real-world instances containing up to 193 transportation requests. Instances with up to 87 customers are solved to optimality within a computation time of one hour.

© 2013 Elsevier B.V. All rights reserved.

The Pickup and Delivery Problem (PDP) consists in defining a set of minimum cost routes to serve a set of independent transportation requests. Each transportation request is defined by an origin called the *pickup point*, a destination called the *delivery point* and a positive value called demand or weight. The PDP is one of the many extensions of the Vehicle Routing Problem (VRP). If time windows for requests are considered, the problem is called the Pickup and Delivery Problem with Time Windows (PDPTW). The demand may be static (each request is known in advance) or dynamic (new requests may arrive at any time). Requests can correspond to the transportation of goods or people. Depending on the context of the application, various objective functions may be considered: number of vehicles used, total ride time, total distance, cost or quality of service-related criteria. There is an abundant literature on pickup and delivery problems and many references can be found in recent surveys by Parragh, Doerner, and Hartl (2006), Berbeglia, Cordeau, Gribkovskaia, and Laporte (2007) and Cordeau, Laporte, and Ropke (2008).

The Pickup and Delivery Problem with Transfers (PDPT) is a recent extension of the PDP relaxing the constraint that the pickup and delivery points of a given request are serviced by the same

vehicle. Instead, some requests may be transferred from one vehicle to another at some predetermined intermediate locations called *transfer points*. Then, distinct vehicles pick up the transferred requests and drives them to their respective delivery point. The PDPT implies a new precedence constraint stating that, if a request uses a transfer point, it must be delivered to the transfer point by a first vehicle before being picked up by a second vehicle. The objective function of the PDPT can be the same as that of the PDP.

Since the PDPT introduces hard precedence and synchronization constraints between vehicles, it is much more difficult to solve in practice than the PDP. The exact methods described in the literature can only solve small instances of the PDPT: Cortés, Matamala, and Contardo (2010) report optimal results for instances with 6 requests, Kerivin, Lacroix, Mahjoub, and Quilliot (2008) report optimal results for instances with 15 requests.

In this paper we introduce the Pickup and Delivery Problem with Shuttle routes (PDPS) which is a special case of the PDPT. The PDPS is a problem that arises in passenger transportation when passengers are picked up at a multitude of locations and dropped off at a few, common delivery points. The set of vehicle routes is split into two categories: *pickup routes* and *shuttle routes*. Pickup routes visit a set of pickup points independently of their associated delivery points and end at one delivery point. Shuttle routes are direct trips between two delivery points. Requests can then be transported in one leg (pickup route) or two legs (pickup route + shuttle route) from their pickup point to their delivery point.

* Corresponding author at: Ecole Polytechnique de Montréal, Département Mathématiques et Génie Industriel, 2900 Boulevard Edouard-Montpetit Montréal, QC H3T 1J4, Canada.

E-mail addresses: renaud.masson@emn.fr (R. Masson), sr@transport.dtu.dk (S. Ropke), fabien.lehuede@emn.fr (F. Lehuédé), olivier.peton@emn.fr (O. Péton).

The main contribution of the paper is to introduce the PDPS, to propose an exact algorithm for this problem based on the branch-and-cut-and-price paradigm and to optimally solve realistic-size instances (up to 87 requests) of the PDPT for the first time, albeit a special case.

The remainder of this paper is organized as follows. Section 1 shows a simple example and discusses the potential benefits of transfers. Section 2 summarizes the related works and positions the present paper in the literature. Section 3 formulates the PDPS, introduces a graph modeling of transfer points, and includes one arc-based and two path-based integer linear programming formulations of the problem. Section 4 describes the branch-and-cut-and-price method used to solve the PDPS. Section 5 presents computational experiments on generated instances with 15–75 requests and on real-life instances with 81–193 requests. Section 6 concludes the paper.

1. Benefits of transfers and motivation for the work

One reason to consider the transfer of requests is the potential savings that can be achieved by this practice. This is illustrated by the example given in Fig. 1. We consider 9 requests: requests 1, 2 and 3 share the pickup point p_{123} , requests 4, 5 and 6 share the pickup point p_{456} , requests 7, 8 and 9 share the pickup point p_{789} . Requests 1, 2, 4, 5, 7 and 8 must be delivered to d_{124578} while requests 3, 6, and 9 must be delivered to d_{369} . The nodes o and o' represent the starting depot and the ending depot respectively. The distances between the points are reported in Fig. 1.

We consider that vehicles have a capacity of 3 and assume that routes are doubly-open, meaning that they start at first pickup point and end at the last delivery point. The optimal solution without any transfers uses three vehicles: vehicle 1 picks up requests 1, 2 and 3 at point p_{123} then unloads requests 1 and 2 at d_{124578} and request 3 at d_{369} . Vehicle 2 starts at p_{456} , visits d_{124578} and ends at d_{369} . Vehicle 3 starts at p_{789} , visits d_{124578} and ends at d_{369} . Each route has a cost of 4 so the total routing cost is 12.

If the transfer of passengers is allowed at point d_1 , vehicles 2 and 3 can be totally emptied and end their trip at d_1 . Then requests 6 and 9 can be loaded into vehicle 1 with request 3. The vehicle follows its itinerary to point d_2 . Vehicle 1 has a route cost of 4 while vehicles 2 and 3 have a route cost of 2 so the total routing cost is 8. In this case, the use of a transfer point enables 33% to be saved on the cost of the optimal solution.

However, transfers also have negative impacts which can put the brakes on the practical implementation of such a system. The main drawbacks are the cost of building or running the facility corresponding to the transfer point, the need for tight synchronization between vehicles and the supplementary handling operations at the transfer point. Moreover, in the context of demand-responsive transportation, direct trips are often preferred by passengers. Transfers are seen as a reduction of the quality of the service. Thus,

the decision makers have to balance the expected benefits and drawbacks of transfers.

2. Related works and position of the problem

Although the PDP has been intensively studied in the last 15 years, the PDPT has been the subject of very few works. Cortés et al. (2010) present a mathematical formulation of the PDPT that is solved using a branch-and-cut algorithm. Instances with 6 requests and 2 vehicles are solved to optimality. Kerivin et al. (2008) consider a PDP where every request can be split as well as transferred from one vehicle to another at every node of the problem. This problem has no time window and is solved using branch-and-cut. Some instances with up to 15 requests are solved to optimality. Nakao and Nagamochi (2010) calculate a lower bound for the PDPT with a single transfer point. Transfers also appear in the school bus model of Fugenschuh (2009). In this work, the routes are already designed and the objective is to minimize the number of buses needed to cover these routes, allowing some transfer between buses.

As far as heuristic approaches are concerned, Mitrović-Minić and Laporte (2006) propose a local search method for the uncapacitated PDPT with a Manhattan distance. They solve generated instances with 100 requests. An Adaptive Large Neighborhood Search (ALNS) for the PDPT was proposed by Masson, Lehuédé, and Péton (2013a) and applied to the instances of Mitrović-Minić and Laporte (2006) and real-world instances with between 2 and 33 transfer points and between 55 and 193 requests. This method was adapted to the Dial-A-Ride Problem with Transfers in Masson, Lehuédé, and Péton (in press). In this study, the authors underline the complexity of enforcing the maximum ride time constraint.

Petersen and Ropke (2011) present another ALNS to handle freight transportation problems considering one transfer point and up to 982 requests. Qu and Bard (2012) propose a GRASP with an ALNS for solving generated instances with 25 requests and one transfer point. Their method can solve 88% of the instances with a distance no greater than 1% of the optimum.

The use of transfer points has also been considered, with special restrictions, for solving pickup and delivery problems. Russell, Morrel, and Haddox (1986) present a school bus routing application where some children are brought to a transfer point, where they are transferred into shuttle buses that take them to their school. The authors present a heuristic to solve this problem. In this approach, the design of the shuttle routes is carried out as a post-processing step. Shang and Cuff (1996) consider that any point can be used as a transfer point. Transfers are only considered to insert a request that cannot be inserted in the current solution without resorting to an extra vehicle. The heuristic relies on the construction of mini-routes which are assigned to vehicles. Thangiah, Fergany, and Awam (2007) use the same principles in a real-time version of the problem. Lin (2008) presents a PDP with time windows where all requests share the same delivery point but have distinct delivery time windows. In this problem, it is considered that a transfer can occur on the last pickup before a delivery. The author uses a heuristic based on a set-partitioning formulation of the problem to solve instances with up to 100 requests. Oertel (2000) presents a tabu search for a version of the problem with two transfer points. Gørtz, Nagarajan, and Ravi (2009) consider a version of the PDPT where the objective is to minimize the makespan. Approximation algorithms are proposed to solve the uncapacitated and capacitated cases. A heuristic column generation method is proposed by Mues and Pickl (2005). They consider a problem with a single transfer point through which requests are systematically routed. The set of routes is composed of pickup routes ending at the transfer point and delivery routes starting at this transfer point. Instances with up to 70 requests are considered for experiments.

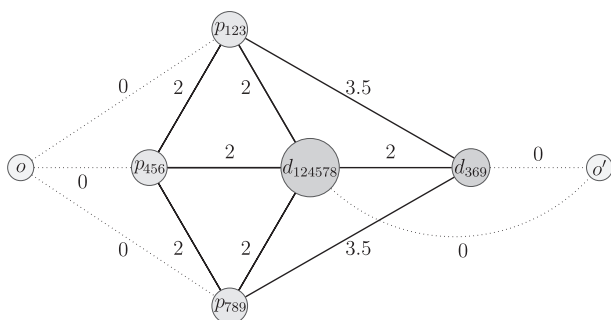


Fig. 1. Example where using transfer point leads to significant savings.

3. The pickup and delivery problem with shuttle routes

This section formally presents three mathematical models for the PDPS. The PDPS was motivated by applications in the field of transportation of people with disabilities (Lehuédé, Pavageau, & Péton, 2009). This concerns people who require daily trips to and from their home to schools or vocational rehabilitation centers. The centers or local authorities have to pay for the increasing transportation costs. In practice, dozens of people share the same delivery point so the number of pickup points is significantly larger than the number of delivery points. In order to reduce transportation costs, some centers resort to transfers, which involves transferring some users from one vehicle to another at appropriate locations (generally some centers). People living in the same geographical area are picked up by one vehicle and transported to a center, where people going to the same destination are gathered in a single vehicle.

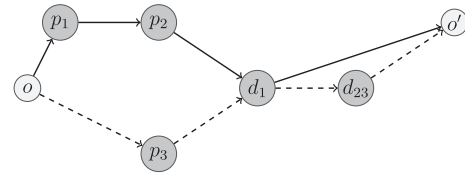


Fig. 2. Example of a physical network.

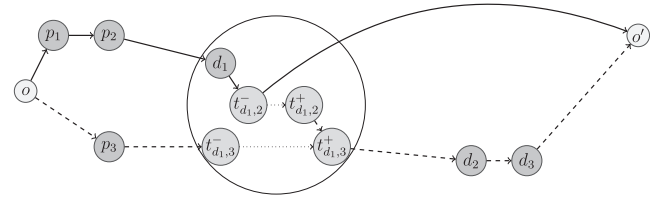


Fig. 3. Logical representation of the physical network of Fig. 2.

3.1. Problem statement

We assume that the time window at a delivery point is the same for each request asking for a transportation up to this point. This time window is determined by the opening hour of the school that the passengers are brought to. The distance between delivery points and tight time windows constraints at the delivery points imply that it is realistic to assume that any vehicle at most can serve two delivery points in the same trip.

A second simplifying hypothesis is that a vehicle does not service a pickup point between two delivery points. The transfer points are positioned at the delivery points, because there typically are staff available there at no additional cost. Several requests are generally transferred or delivered at delivery points; we consider that the time of service at delivery points is a constant which is independent of the number of requests getting in or off the vehicle.

We consider a homogeneous fleet of vehicles K with capacity Q . Each vehicle route starts at an initial depot denoted by o , visits a set of pickup points, goes to a delivery point where some passengers are dropped off and some other are picked up, coming from another vehicle. Then, the vehicle heads for a second delivery point where all passengers are dropped off. The vehicle terminates its route at the end depot o' . Note that some vehicles may not visit any pickup point, but carry shuttle routes between two delivery points, whereas some other vehicles may return to the depot after the first delivery point. In the following we consider open routes, that is, all distances to and from the two terminal points o and o' are assumed to be zero. The objective of the problem is to minimize the total distance traveled by the vehicle fleet.

3.2. Modeling of transfer points

The set T of transfer points is decomposed into a set T^- of inbound transfer nodes where requests get off vehicles and a set T^+ of outbound transfer nodes where requests are reloaded into vehicles. Figs. 2 and 3 detail such a decomposition. Fig. 2 gives an example of a physical network covered by two vehicles. Vehicle 1 picks up requests 1 and 2 and transfers request 2 at point d_1 , where request 1 is delivered. A second vehicle picks up request 3, then loads request 2 before delivering requests 2 and 3 at a common delivery point d_{23} .

The equivalent logical representation is given in Fig. 3. The transfer point d_1 is decomposed into two inbound transfer nodes $t_{d_1,2}^-$, $t_{d_1,3}^-$ and two outbound transfer nodes $t_{d_1,2}^+$, $t_{d_1,3}^+$. In practice all these nodes share the same physical location represented by a circle. The transfer of request 2 is represented by the dotted arc between $t_{d_1,2}^-$ and $t_{d_1,2}^+$.

In the remainder of this article the term *node* refers to the logical representation of the physical network. Let P define the set of pickup nodes and D the set of delivery nodes. Thus, the set N of all nodes of the problem is $N = P \cup D \cup T^- \cup T^+ \cup \{o\} \cup \{o'\}$. The set of points that can be used as delivery or transfer points is denoted by T . We consider a set R of requests. Each request $r \in R$ has a pickup node p_r and a delivery node d_r . Without loss of generality, we assume that all pickup nodes are distinct. Several requests generally share the same physical delivery location; the delivery location of request $r \in R$ is denoted by $b_r \in T$. If two requests r and r' have the same physical delivery location, we split it into two logical vertices d_r and $d_{r'}$ in the logical representation.

We define time windows $[e_i, l_i]$ at node $i \in N$. $t_{\tau,r}^-$ and $t_{\tau,r}^+$ are the inbound and outbound transfer nodes of request $r \in R$ at transfer point $\tau \in T$. q_i is the number of customers loaded/unloaded at node $i \in N$. This number may be negative if customers are unloaded at this node.

3.3. Arc-based formulation

The set of arcs between the nodes of the problem is denoted by A . We write $\delta^-(i) \subset A$ for the set of arcs entering node $i \in N$ and $\delta^+(i) \subset A$ for the set of arcs leaving node $i \in N$. The traversal of each arc $a \in A$ is associated with a known travel time θ_a and cost c_a .

The set of arcs considered includes all arcs that potentially can be traversed by vehicles. Arcs that are not consistent with the definition of the problem are not part of A . In particular: there is no arc from a transfer node or a delivery node to a pickup node. There is no arc from an inbound transfer node, a delivery node or a pickup node to an outbound transfer node. Arcs between inbound transfer nodes located at distinct transfer points are also not part of A . There is no arc between an inbound transfer node and a delivery node located at another delivery point. Symmetrically, arcs between a delivery node and an inbound transfer node located at another transfer point are not part of A . There is no arc between outbound transfer nodes located at distinct transfer points and no arc between two delivery nodes located at distinct delivery points. There is also no arc entering the starting depot or leaving the ending depot.

Some additional arcs are excluded from A in order to break some symmetries and reinforce the linear relaxation of the model. The details about these excluded arcs are given in the Appendix. In the remainder of the article, without loss of generality, we consider that service durations at nodes are taken into account in the matrix θ of travel times.

We define the following decision variables: $z_a^k = 1$ if arc $a \in A$ is traversed by vehicle $k \in K$ and 0 otherwise, u_i^k is the load of vehicle $k \in K$ before servicing node $i \in N$, s_i^k is the time at which vehicle $k \in K$ services node $i \in N$.

The arc-based formulation **AB** of the PDPS can then be formulated as follows:

AB

$$\min \sum_{a \in A, k \in K} z_a^k c_a \quad (1)$$

$$\text{s.t.} \sum_{a \in \delta^-(i), k \in K} z_a^k = 1 \quad \forall i \in P \cup D \quad (2)$$

$$\sum_{a \in \delta^-(i)} z_a^k = \sum_{a \in \delta^+(i)} z_a^k \quad \forall i \in P \cup D \cup T^- \cup T^+, k \in K \quad (3)$$

$$\sum_{a \in \delta^+(o)} z_a^k = 1 \quad \forall k \in K \quad (4)$$

$$\sum_{a \in \delta^-(o')} z_a^k = 1 \quad \forall k \in K \quad (5)$$

$$\sum_{a \in \delta^+(p_r)} z_a^k = \sum_{a \in \delta^+(d_r)} z_a^k + \sum_{\tau \in T, a \in \delta^+(t_{r,\tau})} z_a^k \quad \forall r \in R, k \in K \quad (6)$$

$$\sum_{a \in \delta^+(d_r)} z_a^k = \sum_{a \in \delta^+(p_r)} z_a^k + \sum_{\tau \in T, a \in \delta^+(t_{r,\tau})} z_a^k \quad \forall r \in R, k \in K \quad (7)$$

$$\sum_{a \in \delta^+(t_{r,\tau}), k \in K} z_a^k = \sum_{a \in \delta^+(t_{r,\tau}), k \in K} z_a^k \quad \forall r \in R, \tau \in T \quad (8)$$

$$s_j^k - s_i^k - \theta_a \geq M(z_a^k - 1) \quad \forall i \in N, j \in N, k \in K, a \in \delta^+(i) \cap \delta^-(j) \quad (9)$$

$$u_j^k - u_i^k - q_i \geq M(z_a^k - 1) \quad \forall i \in N, j \in N, k \in K, a \in \delta^+(i) \cap \delta^-(j) \quad (10)$$

$$e_i \leq s_i^k \leq l_i \quad \forall i \in N, k \in K \quad (11)$$

$$0 \leq u_i^k \leq Q \quad \forall i \in N, k \in K \quad (12)$$

$$z_a^k \in \{0, 1\} \quad \forall a \in A, k \in K \quad (13)$$

The objective function (1) sums the costs of all arcs traversed by the vehicles. Constraints (2) state that each pickup and each delivery node should be serviced exactly once. Constraints (3) are the flow conservation constraints. Constraints (4) and (5) state that each vehicle has to leave the starting depot and enter the ending depot exactly once. Unused vehicles follow arc (o, o') at cost 0. Constraints (6) state that if a pickup is serviced on a route, then the corresponding delivery node or an inbound transfer node associated with this request is served on this route. Constraints (7) are similar to constraints (6): if a delivery node is served on a route, then either the pickup node or an outbound transfer node associated with this request is served on this route. Constraints (8) impose that if a given inbound transfer node is reached by one route then the associated outbound transfer node is served by one of the other routes. Constraints (9) state that if a node j is serviced after a node i in a route, the time of service at j is greater than the time of the beginning of service at i plus the service duration and the travel time between the nodes. Using the same reasoning as (9) for travel time, (10) model load conservation constraints. Constraints (11) are time window constraints and constraints (12) are vehicle capacity constraints.

It should be noted that the size of model **AB** grows with the size of the instances. The size of the logical representation can be computed in the following way. The number of vertices in the logical representation of model **AB** is $2|R||T| + 2$. This number of vertices is decomposed in the following way: 2 vertices representing the starting and ending depots, and for each request $r \in R$, 1 pickup point, 1 delivery point, $|T| - 1$ inbound transfer points and $|T| - 1$ outbound transfer points. The exact number of arcs actually depends on the constraints of the problem. However an upper bound

on the number of arcs can be computed in the following way: there are $|K||R|$ arcs from the depots to a pickup node. There are at most $|K||R|(|R| - 1)$ arcs from one pickup node to another; $|K||R|^2|T|$ arcs from a pickup node to a delivery/inbound transfer node; $|K||R||T|(|R| - 1)$ arcs from an inbound transfer node to another; $|K||R||T|(|R| - 1)$ arcs from an inbound transfer node to a delivery node; $|K||R||T|(|R| - 1)$ arcs from an outbound transfer node to another; $|K||R|^2|T|(|R| - 1)$ arcs from an outbound transfer node to a delivery node; $|K||R||T|(|R| - 1)$ arcs from a delivery node to another; $|K||R|$ arcs from a delivery node to the depot. This sums up to a maximum of $|K|(|R|^2 + |R| + 4|R|^2|T| - 4|R||T| + |R|^2|T|^2)$ arcs.

3.4. Computational complexity of the pickup and delivery problem with shuttle routes

The PDPS is NP-hard since any instance of the the vehicle routing problem with time windows (VRPTW) can be transformed to a PDPS instance in polynomial time and since the VRPTW is known to be NP-hard (see e.g. Toth & Vigo (2002)). A VRPTW instance is transformed to a PDPS instance as follows: each VRPTW customer gives rise to a request in the PDPS with pickup at the location of the VRPTW customer and delivery at the depot which also represents the only transfer point. The demand of the request is set equal to the demand of corresponding VRPTW customer and vehicle capacity is replicated from the VRPTW instance. Time windows on pickup nodes are copied from the corresponding VRPTW customers while time windows on delivery nodes can be set such that it is impossible to visit a pickup node after performing a drop-off.

3.5. Path-based formulations

This section introduces two set partitioning formulations **SP1** and **SP2** based on paths. Path-based formulations are known to yield stronger LP relaxation than arc-based formulations. On the other hand, they introduce a huge number of decision variables. The column generation technique used to solve such models is detailed in Section 4.1.

In the path-based models we split routes into two independent legs on each side of the transfer point.

Definition 1. A pickup route starts at o , visits a sequence of pickup points, is emptied at one transfer or delivery point and returns to o' . A shuttle route starts at o , loads a set of requests at one transfer point, then unloads all these requests at the next stop before returning to o' .

This definition is consistent because the distance to the depot is zero and there are always vehicles available. In practice, a pickup route directly followed by a shuttle route can be performed by the same vehicle. However, these routes are distinct in the model.

A path is an ordered sequence of nodes between o and o' , such that every arc traversed is part of A , vehicle capacity Q and time windows constraints are satisfied. The cost of a path is the sum of the costs of its arcs.

We call Ω the set of all paths and Ω_p the set of pickup paths. The cost of path $p \in \Omega$ is written c_p , $a_{i,\tau}^p$ is a binary value equal to 1 if the node $i \in P \cup T^+$ and the delivery point $\tau \in T$ are both serviced by path $p \in \Omega$. $v_{\tau,\tau'}$ is the cost of routing a vehicle from the delivery point $\tau \in T$ to delivery point $\tau' \in T$. Finally, R_τ is the set of requests $r \in R$ delivered at point $\tau \in T$ (i.e. such that $b_r = \tau$). The use of the binary data $a_{i,\tau}^p$ implies a different graph model: each pickup point is duplicated in as many nodes as there are delivery points. If a pickup route visits a pickup point p_i and terminates at delivery d_j , then it serves the duplicated node denoted $p_i^{d_j}$ in this representation.

Fig. 4 illustrates this new graph representation for the routes presented in Fig. 2. This model is based on the following observa-

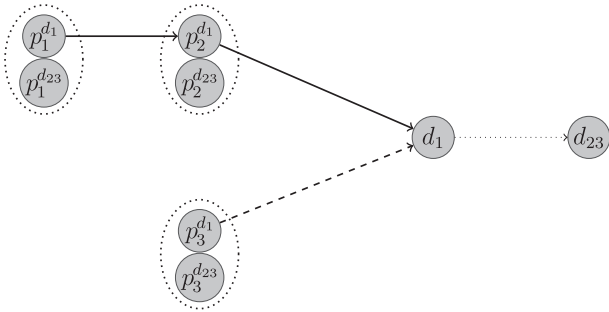


Fig. 4. Graph representation of a solution in path-based formulations.

tions: in Fig. 2, it is not necessary to identify which vehicle performs the shuttle route between d_1 and d_{23} to calculate the cost of the solution (this cost is the same if the plain route or the dashed route serves d_{23} after d_1). In addition, arcs from and to the depot have cost zero, so they do not need to be included in routes. In Fig. 2, there are two delivery points (d_1 and d_{23}), therefore for each pickup point we define two nodes in the graph representation. The plain route picks up request 1 and 2 and terminates at d_1 . This is represented by nodes $p_1^{d_1}$ and $p_2^{d_1}$. The dashed route sequentially visits p_3 , d_1 and d_{23} . In Fig. 4, we attribute only the pickup part of this route to a vehicle. This part ends at delivery point d_1 , therefore its visit at p_3 is modeled by the service of node $p_3^{d_1}$. The dotted arc models one shuttle route from d_1 to d_{23} . In particular, this model is used to define dominance rules in the column generation approach proposed in Section 4.1.

The decision variables are the following:

$\omega_p = 1$ if route $p \in \Omega$ is used in the solution and 0 otherwise, $y_{\tau,\tau'}$ is an integer variable representing the number of vehicles performing shuttle routes between the delivery points $\tau \in T$ and $\tau' \in T$. We consider $y_{\tau,\tau} = 0, \forall \tau \in T$.

Below we introduce models **SP1** and **SP2**. Both models aim to select a set of paths which service all requests at minimum cost. Model **SP1** separates the cost of pickup and shuttle routes and ignores which requests are transported in each shuttle route. Since the fleet is homogeneous, knowing the number of shuttle routes between each pair of delivery points suffices to compute the cost of a solution. Knowing which request is served by which vehicle can easily be computed in a post-processing step. In model **SP2**, the requests transported by each shuttle route are explicitly determined. The mathematical modeling does not distinguish shuttle routes from pickup routes.

On the one hand, the linear relaxation of model **SP1** should be faster to solve, as the shuttle routes are directly managed in the master problem so fewer columns need to be generated. On the other hand, the linear relaxation of model **SP2** provides better lower bounds.

3.6. Model SP1 – the master problem

In model **SP1**, the exact composition of the vehicles following a shuttle route between delivery points is not computed. Model **SP1** can be stated as follows:

SP1

$$\min \sum_{p \in \Omega_p} \omega_p c_p + \sum_{\tau \in T, \tau' \in T} y_{\tau,\tau'} v_{\tau,\tau'} \quad (14)$$

$$\text{s.t.} \sum_{p \in \Omega, \tau \in T} \omega_p a_{p,\tau}^p = 1 \quad \forall i \in P \quad (15)$$

$$\sum_{p \in \Omega, r \in R_r} \omega_p a_{p_r,\tau}^p a_{p_r} \leq Q y_{\tau,\tau'} \quad \forall \tau \in T, \tau' \in T | \tau' \neq \tau \quad (16)$$

$$\omega_p \in \{0, 1\} \quad \forall p \in \Omega_p, \quad (17)$$

$$y_{\tau,\tau'} \in \mathbb{Z}^+ \quad \forall \tau \in T, \tau' \in T \quad (18)$$

The objective function (14) is divided into two parts representing the cost of pickup routes and shuttle routes respectively. Constraints (15) force each pickup node to be serviced exactly once. Constraints (16) ensure that the number of vehicles going from delivery point τ to delivery point τ' is sufficient to transport all requests from $\tau \in T$ to $\tau' \in T$.

3.7. Model SP2 – the master problem

In model **SP2**, the composition of the shuttle routes is formulated explicitly.

SP2

$$\min \sum_{p \in \Omega} \omega_p c_p \quad (19)$$

$$\text{s.t.} \sum_{p \in \Omega, \tau \in T} \omega_p a_{p,\tau}^p = 1 \quad \forall i \in P \quad (20)$$

$$\sum_{p \in \Omega} \omega_p a_{p_r,\tau}^p = \sum_{p \in \Omega} \omega_p a_{p_r,\tau'}^p \quad \forall \tau \in T, \tau' \in T, r \in R_r | \tau' \neq \tau \quad (21)$$

$$\omega_p \in \{0, 1\} \quad \forall p \in \Omega \quad (22)$$

The objective function (19) is the sum of all routes costs. Constraints (20) impose that each pickup node is serviced exactly once. Constraints (21) ensure that each request arriving at a transfer point is transferred to a vehicle that brings it to its delivery point.

4. A branch-and-cut-and-price algorithm

The branch-and-cut-and-price algorithm consists in using column generation to compute the linear relaxation of each node of the branch-and-cut tree. This method has already been successfully applied to solve the related Pickup and Delivery Problem with Time Windows (Ropke & Cordeau, 2009; Baldacci, Bartolini, & Mingozzi, 2011). For each node, as long as valid inequalities strengthening the linear relaxation are found, they are added. This section describes algorithms for solving the pricing sub-problem associated with **SP1** and **SP2**, it introduces three families of valid inequalities used for both path-based formulations of the PDPS and it presents the branching rules that are used to eliminate fractional solutions.

4.1. Column generation for the PDPS

Path-based formulations typically include millions or billions of decision variables and cannot be handled directly. The general idea of column generation is to solve a linear relaxation which only considers a small subset of decision variables. If the result of the restricted problem is optimal, the algorithm stops. To check the optimality of a solution, a sub-problem called a pricing problem, is solved to identify columns having a negative reduced cost. If such columns are identified, they are appended to the subset of columns considered, and the problem is solved again. The process goes on for as long as the sub-problem finds columns with a negative reduced cost.

In the following, we introduce two sub-problems. The first aims to identify columns representing pickup routes, while the second searches for shuttle routes.

4.1.1. Designing pickup routes: an ESPPRC approach

Generating pickup routes amounts to solving an Elementary Shortest Path Problem with Resource Constraints (ESPPRC). The ESPPRC aims to generate an elementary shortest path in a graph that respects some constraints on the consumption of resources (Feillet, Dejax, Gendreau, & Guéguen, 2004; Irnich & Desaulniers,

2005, chap. 2). In our case, we seek a shortest path between o and o' where the resources considered are the capacity of the vehicles and the time of service at the nodes.

The algorithm used to solve the ESPPRC builds partial paths, which are ordered sequences of nodes that begin at the starting depot and return to the ending depot. A feasible extension of a partial path is a path that strictly includes this partial path without violating any constraint of the problem. The solution of the ESPPRC relies on a labeling principle which considers every feasible extension of a partial path that can lead to an optimal solution. In the construction of the path, the time of service at nodes is scheduled as early as possible.

This sub-problem is solved repeatedly $|T|$ times. At each repetition, a distinct delivery point is selected to be the last point visited before the ending depot. We only consider requests that should be delivered at this point or can be transferred at this point without any time window violation. Knowing the final destination of the route, we can easily ensure that a partial path has a feasible extension. It consists in checking that every request already serviced by the route can arrive on time at its delivery node.

In order to solve this sub-problem, a label-correcting algorithm is used as proposed by Feillet et al. (2004). Each label represents a partial path. At each iteration, all the labels representing partial paths ending at a given node i are extended to the nodes that can be reached from i without violating any constraint of the problem. When a new label is generated, it is checked against the set of existing labels, in order to determine if it is dominated by a label, or if it dominates some labels. The labels that are dominated are discarded. When no label remains to be extended, the optimal solution has been identified.

The ESPPRC labeling algorithm relies on labels to model partial paths from o to any node i . A label indicates which is the current node as well as the consumption of resources along the partial path. It is modeled as a 6-tuple of the form (n, t, \bar{c}, l, m, S) . If \mathcal{A} is a label then $n(\mathcal{A})$ denotes the last node of the partial path represented by \mathcal{A} . $t(\mathcal{A})$ is the time of service at node $n(\mathcal{A})$. $\bar{c}(\mathcal{A})$ represents the reduced cost of the partial path represented by \mathcal{A} . $l(\mathcal{A})$ denotes the load of the vehicle after servicing node $n(\mathcal{A})$. $m(\mathcal{A})$ is the latest time at which the delivery node can be serviced without leading to a time window violation for the requests serviced by the partial path. $S(\mathcal{A})$ represents the set of requests that can still be serviced. A request $r \in R$ can be serviced if its pickup node can be serviced from $n(\mathcal{A})$ within its time window and its delivery node can be serviced within its time window and before $m(\mathcal{A})$.

Let us denote $\pi_i \in \mathbb{R}$ the dual variable associated with constraints (15) and (20) in formulations **SP1** and **SP2**, $\lambda_{\tau, \tau'} \in \mathbb{R}^-$, the dual variables associated with constraints (16) in model **SP1** and $\mu_{r, \tau} \in \mathbb{R}^-$ the dual variables associated with constraints (21) in model **SP2**. In model **SP1** the reduced cost of a path p going to point $\tau \in T$ is

$$\bar{c}_p = c_p - \sum_{i \in P} a_{i, \tau}^p \pi_i - \sum_{\tau' \in T \setminus \{\tau\}, r \in R_{\tau'}} a_{p, \tau}^p \lambda_{\tau, \tau'}$$

In model **SP2** the reduced cost of a path p going to point $\tau \in T$ is

$$\bar{c}_p = c_p - \sum_{i \in P} a_{i, \tau}^p \pi_i - \sum_{\tau' \in T \setminus \{\tau\}, r \in R_{\tau'}} a_{p, \tau}^p \mu_{r, \tau}$$

In the label-correcting algorithm, a dominance criterion is used during the search to discard labels. It can be said that a label \mathcal{A} dominates a label \mathcal{A}' when the reduced cost of any feasible extension of label \mathcal{A}' is larger than the lowest reduced cost of a feasible extension of label \mathcal{A} . When a label is dominated it can be deleted.

Proposition 1 (Dominance criterion). Let \mathcal{A} and \mathcal{A}' be two labels representing partial paths from o . Let us denote \succ the relation given by the following set of conditions:

$$\mathcal{A} \succ \mathcal{A}' \iff \begin{cases} n(\mathcal{A}) = n(\mathcal{A}') \\ t(\mathcal{A}) \leq t(\mathcal{A}') \\ \bar{c}(\mathcal{A}) \leq \bar{c}(\mathcal{A}') \\ l(\mathcal{A}) \leq l(\mathcal{A}') \\ m(\mathcal{A}) \geq m(\mathcal{A}') \\ S(\mathcal{A}') \subset S(\mathcal{A}) \end{cases} \quad (23)$$

If $\mathcal{A} \succ \mathcal{A}'$, then \mathcal{A} dominates \mathcal{A}' .

Proof. Assume that $\mathcal{A} \succ \mathcal{A}'$. Then $S(\mathcal{A}') \subset S(\mathcal{A})$ and any node that can extend \mathcal{A}' can also extend \mathcal{A} . Since $t(\mathcal{A}) \leq t(\mathcal{A}')$, any node that can be serviced after $n(\mathcal{A}')$ without breaking time window constraints can also be serviced after $n(\mathcal{A})$, earlier or at the same time. Since $m(\mathcal{A}) \geq m(\mathcal{A}')$, any sequence of nodes that can be serviced in path \mathcal{A}' can also be serviced in path \mathcal{A} . This also applies to capacity constraints, as the remaining capacity is greater for path \mathcal{A} . Finally, we have $\bar{c}(\mathcal{A}) \leq \bar{c}(\mathcal{A}')$. The reduced cost of a path depends only on the reduced cost of the traversed arcs, thus any extension of \mathcal{A}' has a greater cost than the same extension of \mathcal{A} . Hence every feasible extension of \mathcal{A}' is a feasible extension of \mathcal{A} and has a larger reduced cost. \square

4.1.2. Designing shuttle routes

In formulation **SP2**, shuttle routes between delivery points have to be generated explicitly for each couple of delivery points. Let τ denote the starting transfer point and τ' the ending delivery point of a shuttle route p . Let R^p be the set of requests delivered at τ' by the shuttle route p . The reduced cost of this route can be expressed by

$$\bar{c}_p = v_{\tau, \tau'} + \sum_{r \in R^p} a_{p, \tau'}^+ \mu_{r, \tau}$$

Let us consider the binary variable v_r which indicates if request $r \in R_{\tau'}$ is transported or not from τ to τ' along this shuttle route. The shuttle route with the minimum reduced cost is characterized by the following problem:

$$\min v_{\tau, \tau'} + \sum_{r \in R_{\tau'}} v_r \mu_{r, \tau} \quad (24)$$

$$\text{s.t. } \sum_{r \in R_{\tau'}} v_r q_{p_r} \leq Q \quad (25)$$

$$v_r \in \{0, 1\} \quad \forall r \in R_{\tau'} \quad (26)$$

The constraint (25) states that the number of requests transported by the shuttle vehicle should not exceed its capacity.

Since $v_{\tau, \tau'}$ is a constant term, the previous model is strictly equivalent to

$$\max \left\{ - \sum_{r \in R_{\tau'}} v_r \mu_{r, \tau} \mid \sum_{r \in R_{\tau'}} v_r \leq Q, v_r \in \{0, 1\}, \forall r \in R_{\tau'} \right\}$$

Since Q is an integer and the v_r are binary variables, the problem is solved optimally by selecting the Q requests with the smallest negative $\mu_{r, \tau}$. If fewer than Q requests have a negative $\mu_{r, \tau}$, all the requests with negative $\mu_{r, \tau}$ are selected. Thus, solving this sub-problem amounts to sorting the requests in $R_{\tau'}$.

4.2. Valid inequalities

We consider three families of valid inequalities: rounded capacity inequalities (Naddef & Rinaldi, 2002, chap. 3), two-path inequalities (Kohl, Desrosiers, Madsen, Solomon, & Soumis, 1999)

and subset-row inequalities (Jepsen, Petersen, Spoorendonk, & Pisinger, 2008). The two first families of valid inequalities are defined on aggregated variables x_a that are computed from the z -variables of the arc-based formulation **AB** as $x_a = \sum_{k \in K} z_a^k, \forall a \in A$. We refer to Kohl et al. (1999) for a description of how to handle inequalities defined on the x_a variables in a branch-and-price algorithm. The last family of inequalities are defined directly on the ω_p variables of the path-based formulations. Jepsen et al. (2008) explains how such inequalities can be handled in a branch-and-price algorithm and a unified treatment of the subject of cutting planes in branch-and-price algorithms can be found in Desaulniers, Desrosiers, and Spoorendonk (2011).

4.2.1. Rounded capacity inequalities

The aim of rounded capacity inequalities (Naddef & Rinaldi, 2002; Laporte, Desrochers, & Nobert, 3; Laporte et al., 1984) is to prevent a set of nodes $S \subset P$ from being serviced by a number of vehicles which is not consistent with the capacity constraints. Let us denote by $\delta^-(S) \subset A$ the set of arcs entering S and let $x(\delta^-(S))$ be the flow along the arcs of the set $\delta^-(S)$ in the optimal solution. If $q(S)$ designates the sum of the load of the nodes in S , then the following inequality is valid: $x(\delta^-(S)) \geq \max(1, \lceil q(S)/Q \rceil)$.

To detect the violated valid inequalities, we use the same separation heuristics as in Lysgaard, Letchford, and Eglese (2004). These heuristics have been made available in the CVRPSEP library (Lysgaard, 2003).

4.2.2. Two-path inequalities

Two-path inequalities (Kohl et al., 1999) prevent a set of nodes $S \subset P$ from being serviced by fewer than two vehicles if no feasible solution with one vehicle exists. When S cannot be serviced by a single route, the inequality: $x(\delta^-(S)) \geq 2$ is valid.

In order to detect whether a set S can be serviced by a single route, one should check if the TSPTW on this set has a feasible solution. In order to find candidate sets S we use a simple greedy heuristic: let x^* be the aggregated variables corresponding to the current fractional solution. For each pickup node $p \in P$, the separation heuristic starts with a set $S' = \{p\}$ and repeatedly adds to the set S' the pickup node which minimizes the increase of the value of $x^*(\delta^-(S'))$. As long as $x^*(\delta^-(S')) < 2$, it is checked whether a feasible solution of the TSPTW with the nodes of S' exists. If not, a new violated valid inequality has been exhibited.

4.2.3. Subset-row inequalities

The third family of valid inequalities is subset-row inequalities introduced by Jepsen et al. (2008), which indicate that two routes which visit a common customer are conflicting. Considering a set $S \subset P$ of pickup nodes, and $k \in \mathbb{N}^*$ such that $0 < k \leq |S|$, the following inequality is valid:

$$\sum_{p \in \Omega} \left[\frac{1}{k} \sum_{i \in S, \tau \in T} a_{i,\tau}^p \right] \omega_p \leq \left\lfloor \frac{|S|}{k} \right\rfloor \tag{27}$$

This means that the number of paths of the optimal solution where $|S| \geq k$ should be lower than $\lfloor \frac{|S|}{k} \rfloor$ (otherwise at least one element of S is serviced by more than one route). We therefore search conflicting columns in the optimal solution of the linear relaxation of the PDPS, with a separation heuristic considering all sets S of three customers and $k = 2$. If the set of columns covering these customers violates constraints (27), the corresponding valid inequality is added to the set-partitioning formulation. These inequalities complicate the pricing sub-problem as each active constraint adds an extra resource to the ESPPRC. We refer to the paper by Jepsen et al. (2008) for further details.

4.3. Branching rules

Branching is performed when the solution of the linear relaxation of the master problem is fractional and no more valid inequalities can be found by the separation heuristics. The first node of the branch-and-bound tree, where no branching has been done, is called the root node. The computation time depends largely on the branching strategy. A very common approach is to give priority to branching decisions which eliminate most cases. We propose three branching rules that are used sequentially.

1. Since variables ω_p associated with the use of columns $p \in \Omega$ take either value 0 or 1, their sum should be an integer value. If not, we branch on the value of this sum and create two branches with constraints $\sum_{p \in \Omega} \omega_p \leq \lfloor \sum_{p \in \Omega} \omega_p \rfloor$ and $\sum_{p \in \Omega} \omega_p \geq \lceil \sum_{p \in \Omega} \omega_p \rceil$ respectively.
2. If $\sum_{p \in \Omega} \omega_p$ is integer, we consider branching on the number of vehicles following shuttle routes between two given delivery points. This corresponds to the y variables in model **SP1** and the ω variables associated with the shuttle path columns in model **SP2**. Let us assume that some variable $y_{\tau,\tau'}$ in the optimal solution has a non-integer value $y'_{\tau,\tau'}$. Then in one branch we add the constraint $y_{\tau,\tau'} \geq \lceil y'_{\tau,\tau'} \rceil$ and in the other branch $y_{\tau,\tau'} \leq \lfloor y'_{\tau,\tau'} \rfloor$. In model **SP2**, we replace $y_{\tau,\tau'}$ by $\sum_{p \in \Omega_{\tau,\tau'}} \omega_p$, where $\Omega_{\tau,\tau'}$ is the set of columns that represent shuttle paths between τ and τ' .
3. If the sum of variables $\omega_p, p \in \Omega$ and the number of shuttle routes are both integers, the algorithm branches on the arcs: if the sum of the ω_p variables using a given arc is not an integer, two branches are created: one branch imposes the use of the arc while the other branch forbids it. We branch first on the arc for which the flow is closest to 0.5.

5. Computational experiments

The algorithm presented in this paper was developed in C++ and run on an i3-530 Ubuntu 10.04 computer. We used IBM Ilog Cplex 12.2 to solve all linear relaxations. The main goal of this section is to evaluate the performance of the algorithm described in the preceding section. We first assess the performance of arc-based and path-based formulations on a set of small-sized instances. Next, we measure the efficiency of various combinations of valid inequalities. Models **SP1** and **SP2** are then compared on a collection of generated instances with 25–75 requests. The potential benefits of using transfer points are discussed in Section 5.6. We conclude the section with the results of instances arising from real-world data.

5.1. Generated instances

In order to assess the efficiency of the methods proposed in this article, we have generated several instances. The methodology for generating those instances has been the following. The points of the problem are located in a 100 km × 100 km square, and the Euclidean distance is used. We consider that vehicles travel at a constant speed of 60 km/hour. We consider 30 minutes wide time windows at delivery points. The time windows at pickup points open one hour before the end of those of the delivery points. The service duration at pickup points is one minute. The service duration at delivery/transfer points is five minutes. The number of requests considered is 15, 20, 25, 50, and 75. The number of distinct delivery points is between 2 and 4. We assume that each delivery point can be used as a transfer point. The capacity of

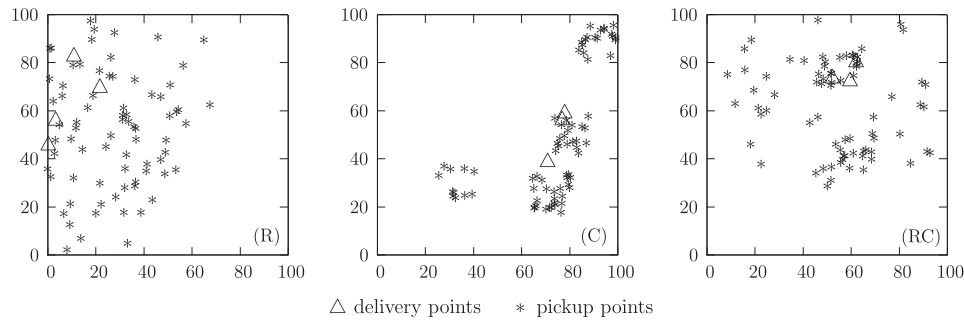


Fig. 5. Location of pickup and delivery points for instances in sets (R), (C) and (RC).

vehicles is 3, 4, 5, 6 or 8 passengers. The purpose of instances with 15 or 20 requests is to compare models **SP1** and **SP2** with model **AB**: 12 instances of each size have been created. These instances have 2 transfer points and vehicle capacities ranging from 3 to 6 passengers. In addition, we defined 27 instances for each of the 25, 50 and 75 requests size. These instances have between 2 and 4 transfer points and vehicle capacity of 4, 6 or 8 passengers. In total, there are 105 instances.

Those instances are classified in three sets based on their geographical distributions of pickup points: *random* (R), *clusters* (C) and *mixed* random-clusters (RC). In the three cases, the position of the first delivery point is selected randomly. Then we iteratively locate other delivery points such that the closest delivery point previously generated is not farther than 25 km. For all instances of set (R), we first randomly select the delivery point of each request. Then the pickup points are randomly selected such that direct trips from the pickup points to the delivery points satisfy the time window constraints. For all instances of set (C), we consider clusters of size $8 \text{ km} \times 8 \text{ km}$. We first randomly choose the centers of the clusters such that at least one delivery point can be reached within 1 hour. Moreover, for each delivery point, at least one cluster must be reached within 1 hour. Then for each request, we randomly select the delivery points and then the pickup points in one of the four clusters such that time window constraints are satisfied. Instances in set (RC) mix the properties of sets (R) and (C) and have 50% of their pickup points belonging to clusters.

Fig. 5 illustrates the distribution of the pickup and delivery points for one instance in each set. These instances are available online at https://subversion.assembla.com/svn/pdps_instances.

5.2. Arc-based formulation vs. path-based formulation

In order to compare the arc-based formulation **AB** and the path-based formulations **SP1** and **SP2**, we use a collection of 24 small-sized instances with 15 and 20 requests, 2 delivery nodes and a capacity of 4, 5 or 6. The size of model **AB** makes it prohibitively hard to solve when the size of instances grows. This is why we only solve instances with 15 and 20 requests.

Table 1 reports the optimality gap obtained when solving the linear relaxations. The optimal solutions are computed using the branch-and-cut-and-price framework described in this paper.

Note that the linear relaxations of model **AB** have been solved with IBM ILOG Cplex 12.2. In the last column of Table 1, the original formulation of model **AB** have been enhanced with the cuts automatically added by Cplex. If these cuts are disabled the average gap is 47.2%. These results show that the linear relaxation of path-based models clearly outperforms that of the arc-based model, even if Cplex cuts significantly improve the lower bound of model **AB**.

Table 1

Solutions of the linear relaxation of the arc-based model and the path-based models. The first column indicates instance sets. The second column is the number of requests considered. The third column is the capacity of the vehicle. The last column shows the gap between the optimal value z^* and the solution of the linear relaxation of model **AB**.

Set	Requests	Q	Gap (SP1) (%)	Gap (SP2) (%)	Gap (AB) (%)
C	15	3	3.27	2.07	36.52
		4	5.43	0.15	50.49
		5	1.96	0.00	34.33
		6	0.00	0.00	18.99
	20	3	4.69	4.66	47.97
		4	3.47	1.41	32.07
		5	6.42	3.12	44.32
		6	4.57	4.57	34.68
R	15	3	6.69	2.14	26.22
		4	3.00	1.31	26.33
		5	4.44	0.00	53.78
		6	4.73	2.79	74.79
	20	3	2.76	0.00	26.05
		4	1.40	0.00	29.15
		5	1.41	0.00	35.99
		6	2.45	1.55	25.13
RC	15	3	2.63	1.90	12.23
		4	8.30	3.02	30.47
		5	1.19	0.00	35.56
		6	0.00	0.00	27.32
	20	3	0.89	0.36	21.09
		4	1.12	0.07	22.33
		5	2.87	1.59	37.03
		6	8.20	1.26	29.26
Average			3.41	1.33	33.84

5.3. Strengthening the linear relaxation through valid inequalities

From now on, we only consider path-based models. We assess the performance of the proposed algorithms on three sets of generated instances with 25, 50 and 75 requests respectively.

Our first numerical experiment concern the use of valid inequalities introduced in Section 4.2. We seek a trade-off between the number of branch-and-bound nodes explorations saved thanks to the cuts and the computation time required to separate these cuts. We run experiments on a subset of 10 representative instances, considering the following 12 settings:

- NoCut:** No valid inequalities.
- RC-root:** Rounded capacity inequalities are generated at the root node.
- RC-all:** Rounded capacity inequalities are generated at every node of the branching tree.
- 2P-root:** Two-path inequalities are generated only at the root node.

Table 2

Solving times (in seconds) for the 12 configurations of valid inequalities with model **SP1** (part 1/2). The first three columns show the characteristics of the instances: set of instances, number of points (# pickup + # delivery) and capacity of vehicles. Columns 4–10 show the time needed to solve the instances to optimality. Time out (t.o.) means that the required CPU was longer than 3600s. Memory overflow cases are reported by –.

Instance	Configuration		Configuration								
			Set	Points	Q	NoCut	RC–root	RC–all	2P–root	2P–all	SR–root
C	25 + 4	4	33	32	44	33	43	50	27		
C	25 + 2	6	2808	2799	5	4	1	3	0		
R	25 + 3	4	1	1	3	2	3	1	2		
C	25 + 4	6	12	12	11	13	12	6	11		
C	25 + 4	8	60	59	102	60	110	33	101		
RC	50 + 2	4	t.o.	570	359	t.o.	353	1391	339		
RC	50 + 3	4	t.o.	t.o.	53	t.o.	72	25	47		
RC	50 + 4	4	–	t.o.	t.o.	–	t.o.	t.o.	t.o.		
C	50 + 3	4	t.o.	100	224	t.o.	262	104	356		
C	50 + 3	6	t.o.	1325	753	t.o.	1240	2835	1206		
Average			2091	1210	515	1811	570	803	571		

Table 3

Solving times (in seconds) for the 12 configurations of valid inequalities with model **SP1** (part 2/2).

Instance	Configuration		Configuration						
			Set	Points	Q	All-root	All-RC	All-2P	All-SR
C	25 + 4	4	27	29	53	28	47		
C	25 + 2	6	6	4	7	1	0		
R	25 + 3	4	1	1	3	1	3		
C	25 + 4	6	6	7	34	3	14		
C	25 + 4	8	34	39	110	53	101		
RC	50 + 2	4	346	315	410	276	364		
RC	50 + 3	4	33	47	91	45	46		
RC	50 + 4	4	t.o.	t.o.	t.o.	t.o.	t.o.		
C	50 + 3	4	137	135	123	185	293		
C	50 + 3	6	813	654	2290	1202	1129		
Average			500	483	672	539	560		

Table 5

Solving times (in seconds) for the 12 configurations of valid inequalities with model **SP2** (part 2/2).

Instances	Configuration		Configurations						
			Set	Points	Q	All-root	All-RC	All-2P	All-SR
C	25 + 4	4	52	45	73	43	71		
C	25 + 2	6	1	1	1	1	1		
R	25 + 3	4	4	3	4	3	4		
C	25 + 4	6	6	5	17	4	17		
C	25 + 4	8	106	85	183	91	152		
RC	50 + 2	4	860	864	1066	948	1028		
RC	50 + 3	4	79	86	88	97	113		
RC	50 + 4	4	t.o.	t.o.	t.o.	t.o.	t.o.		
C	50 + 3	4	417	429	451	590	707		
C	50 + 3	6	952	729	1204	2703	t.o.		
Average			608	585	669	808	929		

- 2P–all:** Two-path inequalities are generated at every node of the branching tree.
- SR–root:** Subset-row inequalities are generated only at the root node.
- SR–all:** Subset-row inequalities are generated at every node of the branching tree.
- All–root:** All valid inequalities are generated only at the root node.
- All–RC:** Cumulates All–root and RC–all.
- All–2P:** Cumulates All–root and 2P–all.
- All–SR:** Cumulates All–root and SR–all.
- All–all:** All inequalities are generated at every node of the branching tree.

Tables 2–5 report the time needed to find the optimal solution for each configuration. The computation is stopped after 1 hour. Tables 2 and 3 report the results for model **SP1**, while Tables 4 and 5 report results for model **SP2**.

Concerning model **SP1**, we observe a factor of at least 3 between configuration NoCut and any configuration using all families of valid inequalities. The best two average running times are achieved by configurations All-Root (500 seconds) and All-RC (483 seconds) while all configurations that include the three types of valid inequalities are among the most efficient. The results also show that the time saved by valid inequalities is offset by the time needed to separate them. This explains why configuration All-Root performs better than configuration All-all. Results for model **SP2**

Table 4

Solving times (in seconds) for the 12 configurations of valid inequalities with model **SP2** (part 1/2).

Instances	Configuration		Configurations								
			Set	Points	Q	NoCut	RC–root	RC–all	2P–root	2P–all	SR–root
C	25 + 4	4	61	61	56	71	76	44	44		
C	25 + 2	6	t.o.	t.o.	t.o.	3	3	1	1		
R	25 + 3	4	4	3	3	4	5	3	3		
C	25 + 4	6	19	19	19	6	17	6	4		
C	25 + 4	8	134	131	137	83	193	958	98		
RC	50 + 2	4	t.o.	t.o.	1940	t.o.	t.o.	t.o.	t.o.		
RC	50 + 3	4	t.o.	t.o.	t.o.	t.o.	1113	60	66		
RC	50 + 4	4	t.o.	t.o.	t.o.	t.o.	t.o.	t.o.	t.o.		
C	50 + 3	4	t.o.	396	585	t.o.	t.o.	424	1274		
C	50 + 3	6	t.o.	t.o.	t.o.	t.o.	t.o.	t.o.	t.o.		
Average			2182	1861	1714	1817	1581	1230	1229		

Table 6

Comparison of running times on instances with and without time windows. The first three columns show the characteristics of the instances: set of instances, number of points (# pickup + # delivery) and vehicles capacity. Columns 4 and 5 report the running times of model **SP1** and **SP2** to solve the instances without time windows. Columns 6 and 7 reports the same information with time windows.

Instances			Without TW		With TW	
Set	Points	Q	SP1	SP2	SP1	SP2
C	25 + 4	4	36	40	29	45
C	25 + 2	6	2	1	4	1
R	25 + 3	4	18	48	1	3
C	25 + 4	6	t.o.	t.o.	7	5
C	25 + 4	8	t.o.	t.o.	39	85
RC	50 + 2	4	7	14	315	864
RC	50 + 3	4	48	222	47	86
RC	50 + 4	4	t.o.	t.o.	t.o.	t.o.
C	50 + 3	4	2012	t.o.	135	429
C	50 + 3	6	t.o.	t.o.	654	729

are similar, the best two average running times are achieved by configurations All-Root (608 seconds) and All-RC (585 seconds). Therefore, in the remainder of this article, we separate all valid inequalities at the root node of the branching tree, and apply RC inequalities on all other nodes.

5.4. Impact of time windows

The next experiments evaluate the impact of time windows on the practical difficulty of the instances. We selected a representative subset of instances for which we created a modified version without time windows. We compare the time necessary to solve models **SP1** and **SP2** with or without time windows. The results are summarized in Table 6.

Table 7

Results of models **SP1** and **SP2** on instances with 25 requests. Columns 1–3 detail the instance set, the capacity of vehicles and the number of transfer points. Column 4 reports the value z^* of the optimal solution, found by both models for instances with 25 requests. Columns 5 and 6 report the lower bound at the root node for both models. The next columns report the run time and the number of nodes explored during the branch-and-cut procedure. In the calculation of average values for columns 7 and 8, the computation times have been rounded up to 1 second for all lines written < 1.

Instance			z^*	Root node		Run time (seconds)		B&C nodes		
Set	Q	D		SP1	SP2	SP1	SP2	SP1	SP2	
R	4	2	554.31	544.35	550.18	<1	<1	6	6	
		3	465.81	442.04	461.25	1	3	51	55	
		4	596.65	580.38	596.65	<1	<1	12	1	
	6	2	552.21	537.49	552.21	<1	<1	2	1	
		3	634.18	609.37	634.18	<1	<1	6	1	
		4	734.50	715.68	734.50	<1	<1	17	1	
	8	2	413.15	395.33	413.15	<1	<1	3	1	
		3	397.13	362.22	397.13	<1	<1	7	1	
		4	617.83	571.88	617.83	1	<1	65	1	
	RC	4	2	395.53	391.28	395.53	<1	<1	3	1
			3	437.48	422.11	437.48	<1	<1	2	1
			4	720.26	706.74	720.26	<1	<1	6	1
6		2	324.66	312.58	324.66	<1	<1	2	1	
		3	449.60	434.16	449.60	<1	<1	14	1	
		4	571.93	555.73	569.27	<1	<1	4	4	
8		2	529.46	516.55	529.46	<1	<1	2	1	
		3	538.09	493.83	538.09	<1	<1	8	1	
		4	452.14	420.27	452.14	1	<1	33	1	
C		4	2	318.81	305.86	318.81	<1	<1	4	1
			3	397.82	393.80	396.90	<1	<1	10	4
			4	323.97	300.15	310.65	29	45	422	381
	6	2	205.31	193.23	205.31	6	<1	4	1	
		3	486.14	477.49	486.14	<1	<1	2	1	
		4	457.22	422.21	432.90	8	5	317	157	
	8	2	223.02	210.12	223.02	1	<1	6	1	
		3	391.04	365.15	382.96	1	3	24	18	
		4	377.81	344.33	364.71	39	85	631	412	
	Average			465.41	445.35	462.78	3.89	6.03	61.59	39.11

Except for a few exceptions it can be noted that instances with time windows are easier to solve. From now on, we consider only instances with time windows.

5.5. Comparison between path-based formulations

We now compare models **SP1** and **SP2** on all generated instances. Tables 7–9 report the running time of each model on instances with 25, 50 and 75 requests respectively. A time limit of 1 hour was used in these experiments.

All instances with 25 requests could be solved to optimality by both models. Table 7 compares the optimal solutions with the lower bound at the root node. It also reports the run times and the number of nodes explored during each run.

All 50 requests instances in set (R) could be solved to optimality. However, only 8 out of 9 instances in set (RC) could be solved by both models within the time limit. Only 7 out of 9 instance in set (RC) could be solved by model **SP1** and 6 of them by model **SP2**. Then, instead of known optimal solutions z^* , Table 8 reports the best solutions obtained, denoted by z_{best} . Whenever z_{best} is not optimal, lower bounds found by **SP1** and **SP2** are also reported.

Table 9 shows that half of the instances with 75 requests could be solved to optimality (14 with model **SP1** and 13 with model **SP2**). We observe that Model **SP2** always gives better lower bounds at the root node. As a result, fewer nodes are then explored during the branch-and-price algorithm. However, when the problem cannot be solved to optimality, the final lower bound provided by both models are very similar.

5.6. Impact of transfers

In order to evaluate the cost savings due to transfers, we apply three distinct scenarios:

Table 8

Results of models **SP1** and **SP2** on instances with 50 requests. Columns 4 and 5 report the value of the best integer solution found by each model. When both model solve the instance to optimality, we report the value between columns 4 and 5. Columns 8 and 9 report the value of the best lower bound obtained before time out. When the instance is solved to optimality, the values in column LB are equal to z_{best} . For readability reasons, we do not report them. However, the average of LB values, calculated in the last line, is the average of all lower bounds, including the ones which are not written explicitly.

Instance			z_{best}		Root node		LB		Run time (seconds)		B&C nodes		
Set	Q	D	SP1	SP2	SP1	SP2	SP1	SP2	SP1	SP2	SP1	SP2	
R	4	2	787.87		780.56	785.28			3	4	7	7	
		3	798.76		783.60	795.44			7	5	58	29	
		4	1033.59		1008.06	1025.67			5	6	69	39	
	6	2	1011.22		984.50	1009.92			1	1	6	4	
		3	1048.53		1012.82	1048.53			1	1	13	1	
		4	1007.64		952.78	1007.64			2	1	21	1	
	8	2	816.99		809.55	814.61			1	2	8	8	
		3	1154.40		1117.77	1154.40			1	1	6	1	
		4	1135.59		1079.66	1131.01			6	2	83	3	
	RC	4	2	619.69		614.97	614.97			318	864	656	621
			3	585.94		572.92	584.15			47	86	74	35
			4	698.56	698.61	660.25	680.12	694.34	691.33	t.o.	t.o.	4440	693
6		2	742.02		738.49	738.49			18	36	19	17	
		3	772.09		739.80	772.09			5	1	10	1	
		4	695.40		665.83	681.93			44	35	135	38	
8		2	649.08		642.04	648.87			81	9	69	2	
		3	702.94		650.89	702			60	5	31	6	
		4	803.17		765.79	801.26			4	4	27	23	
C		4	2	405.28	405.28	398.51	398.57		405.27	2166	t.o.	805	647
			3	454.79		444.82	450.48			132	429	142	128
			4	528.78		518.65	527.29			9	7	4	3
	6	2	393.21	393.21	385.36	385.36	389.11	388.33	t.o.	t.o.	635	171	
		3	488.22		474.67	480.52			652	769	1389	374	
		4	476.90		443.47	476.90			200	6	43	1	
	8	2	450.70	450.70	424.33	425.19	436.73	434.18	t.o.	t.o.	2737	606	
		3	329.81		322.13	327.96			277	487	11	8	
		4	665.81		657.46	662.65			5	7	8	4	
	Average			713.22	713.22	690.73	708.57	712.40	712.16	550	636	426	129

Table 9

Results of models **SP1** and **SP2** on instances with 75 requests.

Instance			z_{best}		Root node		LB		Run time (seconds)		B&C nodes		
Set	Q	D	SP1	SP2	SP1	SP2	SP1	SP2	SP1	SP2	SP1	SP2	
R	4	2	1063.15		1054.45	1061.16			16	19	30	15	
		3	1318.62		1300.37	1305.08			144	214	666	612	
		4	1283.45		1268.14	1278.78			77	184	175	93	
	6	2	1375.73		1375.73	1375.73			t.o.	t.o.	5	4	
		3	1165.62		1133.17	1158.76			13	6	33	18	
		4	1435.46		1398.26	1428.94			18	6	74	10	
	8	2	1381.67		1381.67	1381.67			8	73	6	1	
		3	1382.52		1364.39	1381.53			12	32	41	7	
		4	1596.18		1553.92	1589.21			11	7	51	3	
	RC	4	2	1035.86		1035.03	1035.86			3	2	3	1
			3	1123.09	1126.39	1095.82	1103.56	1115.84	1115.33	t.o.	t.o.	2934	2016
			4	1056.72		1027.83	1055.07			19	9	51	6
6		2	914.59	913.77	908.99	910.62	912.04	912.65	t.o.	t.o.	22	23	
		3	884.46	884.46	870.49	879.96		883.87	1434	t.o.	872	304	
		4	1198.21		1140.40	1197.75			18	6	72	3	
8		2	671.42	677.90	664.54	662.87	664.54	662.87	t.o.	t.o.	1	1	
		3	879.05	881.63	854.06	871.15	878.36	876.39	t.o.	t.o.	2482	624	
		4	1247.60		1231.29	1243			318	591	129	120	
C		4	2	715.56	716.69	697.20	698.72	709.76	709.45	t.o.	t.o.	275	138
			3	669.80	672.95	662.59	662.91	666.59	665.34	t.o.	t.o.	226	41
			4	912.71	886.59	898.67			269	1475	200	618	
	6	2	474.66		471.02	471.42			117	469	6	10	
		3	663.84	664.23	637.79	645.10	649.75	648.80	t.o.	t.o.	83	15	
		4	541.39	540.88	516.50	521.24	529.33	527.74	t.o.	t.o.	82	41	
	8	2	658.88	658.88	645.80	647.81	651.45	651.23	t.o.	t.o.	166	39	
		3	494.26	493.83	454.06	464.05	473.46	464.81	t.o.	t.o.	238	8	
		4	606.55	629.17	574.21	590.63	574.21	590.63	t.o.	t.o.	1	1	
	Average			990.81	992.21	969.88	982.24	986.59	986.57	1558	1712	331	177

- (**PDPS**) We solve the pickup and delivery problem with shuttle routes using model **SP1**.
- (**OVRPTW**) We solve one independent vehicle routing problem with open routes and time windows for each delivery location. In this scenario, the passengers going to distinct delivery locations are not mixed. This is still the most common situation encountered in practice. The algorithm described for the PDPS was adapted for solving the OVRPTW by setting variables $y_{\tau,\tau'}$ at 0.
- (**PDPTW**) We solve a pickup and delivery problem with time windows where each vehicle can visit at most two delivery points and no transfer operation is allowed. We adapted the column generation algorithm by setting variables $y_{\tau,\tau'}$ at 0, and considering a modified ESPPRC where two delivery points can be visited.

The three scenarios were run on the 81 generated instances. With a time limitation of 1 hour, we could solve all scenarios to optimality only for instances with 25 requests. The corresponding results are reported in Table 10.

Table 11 presents the results for all 81 instances. We aggregated the results by the main characteristics of the instances: number of requests, layout of nodes, capacity of the vehicles and number of delivery points. Since the OVRPTW and PDPTW have not been solved up to optimality for every instance, we actually compute a lower bound on savings, by comparing the upper bound of the PDPS to the lower bounds of the OVRPTW and PDPTW.

It seems that the only characteristic having an impact on savings is the number of delivery points in the instance. Thus, the higher the number of delivery points, the higher the savings. However, these savings can vary a great deal from one instance to another. The average values of 3.82% and 2.14% may seem almost negligible but they hide many different cases. The maximal values observed show that, in some cases, significant savings can easily be

Table 10

Impact of transfer points on instances with 25 requests. Columns 1–3 detail the set of instances, the capacity of vehicles and the number of transfer points. Column 4 reports the relative difference between the optimal solution of the PDPS and that of the OVRPTW. The last column reports the relative difference between the optimal solution of the PDPS and that of the PDPTW.

Set	Q	D	PDPS vs. OVRPTW (%)	PDPS vs. PDPTW (%)
R	4	2	-5.32	-1.01
		3	-5.93	-4.85
		4	-5.47	-2.92
	6	2	0.00	0.00
		3	-2.63	0.00
		4	-5.03	-0.69
	8	2	-11.67	-5.14
		3	-6.26	-6.26
		4	-7.12	-5.16
RC	4	2	-1.68	-1.68
		3	0.00	0.00
		4	-3.14	-1.84
	6	2	-2.99	0.00
		3	-9.63	-3.41
		4	-0.06	-0.06
	8	2	-1.22	-1.22
		3	-7.20	-5.57
		4	-5.88	-2.44
C	4	2	-0.94	0.00
		3	-5.75	-1.32
		4	-8.31	-2.77
	6	2	-1.05	0.00
		3	-1.53	-1.53
		4	-0.14	0.00
	8	2	-3.02	-3.02
		3	-0.15	-0.15
		4	-3.09	-3.09

Table 11

Savings due to transfers. The first column is the characteristic considered. Columns 2 and 3 report the average savings compared to the OVRPTW and PDPTW approaches. Columns 4 and 5 report the percentage of instances for which the savings are greater than 5%.

Instances characteristics	Average savings (%)		% of instances with gap > 5%	
	PDPS vs. OVRPTW	PDPS vs. PDPTW	PDPS vs. OVRPTW	PDPS vs. PDPTW
Requests: 25	3.95	1.97	30	7
Requests: 50	3.90	2.49	44	19
Requests: 75	3.59	1.95	26	7
Instance set: R	4.14	2.08	44	19
Instance set: RC	3.57	2.30	30	7
Instance set: C	3.74	2.02	26	7
Q = 4	3.83	2.05	41	11
Q = 6	3.68	1.99	26	7
Q = 8	3.93	2.37	33	15
D = 2	2.13	1.22	7	4
D = 3	4.69	2.55	48	19
D = 4	4.63	2.64	44	11
Average	3.82	2.14	33	11

made thanks to the use of shuttle routes. Identifying the characteristics of instances with a high potential for the use of shuttles remains an interesting topic for further research.

5.7. Application to the transportation of disabled people

In this section we consider real-life data, collected in the area of Nantes, France, concerning people with mental disabilities traveling to specialized schools or vocational rehabilitation centers. The five instances include between 81 and 193 requests and refer to inbound trips from home to the centers. Thus, the pickup nodes are the personal addresses of the passengers and the delivery and transfer points are the centers. All requests have distinct pickup points and correspond to individual demands. The number of centers considered is 2 or 5. The time window $[e_i, l_i]$ for all centers are identical. Values l_i correspond to the official beginning of the daily activity. Values e_i are calculated as $l_i - 25$ minutes: the passengers are allowed to arrive up to 25 minutes prior the official time of opening. The vehicles used are 9 seaters minibuses. Table 12 reports the results of both models.

For instances with 2 centers, the gap between the lower bound and the upper bound is always less than 1.5% and the results of model **SP1** are better on average. The linear relaxation of the largest instance with 5 centers can be solved within one hour only by model **SP1**, with a gap of about 4%. For almost all instances, the algorithm is still adding valid inequalities to the root node at the end of the one-hour time limit. As this is much faster with model

Table 12

Results for real-world instances of model **SP1** and **SP2**. Column 1 is the number of pickup and delivery points in the instances. Columns 2 and 5 show the lower bounds found by models **SP1** and **SP2** respectively. Columns 3 and 6 show the upper bounds found by models **SP1** and **SP2** respectively. Columns 4 and 7 show the gaps between the lower and upper bounds of models **SP1** and **SP2**. “-” means that no upper bound was found after one hour of computation.

P + D	Model SP1			Model SP2		
	LB	UB	Gap (%)	LB	UB	Gap (%)
81 + 2	433.69	436.22	0.58	433.64	436.17	0.58
84 + 2	828.83	835.77	0.83	829.09	836.13	0.84
87 + 2	461.40	461.40	0.00	460.11	463.32	0.69
109 + 2	607.93	614.92	1.14	607.46	616.43	1.45
193 + 5	1098.15	1146.12	4.19	0.00	-	-

Table 13

Savings obtained due to transfers. The first column is the number of requests and the number of delivery points in the instances. Column 2 is the upper bound of the PDPS solution (column 3 of Table 12). Column 3 reports the lower and upper bounds of the PDPTW solution. Column 4 shows the guaranteed savings that can be achieved by the PDPS (the gap between the upper bound of the PDPS and the upper bound of the PDPTW). Column 5 reports the optimal solution of the OVRPTW. Finally column 6 represents the gap between the upper bound of the PDPS and the optimal solution for the OVRPTW.

Points	PDPS (upper bound)	PDPTW		OVRPTW	
		[LB, UB]	Gap (%)	(Opt.)	Gap (%)
81 + 2	436.22	[451.26, 473.23]	3.45	507.25	16.28
84 + 2	835.77	[839.90, 840.30]	0.49	840.30	0.54
87 + 2	461.40	[477.02, 477.25]	3.39	566.03	22.68
109 + 2	614.92	[646.92, 682.80]	5.20	684.83	11.37
193 + 5	1146.12	[1256.01, 1297.92]	9.59	1297.92	13.24

SP1 than with model SP2, this may explain why model SP1 provides better results. Upper bounds are computed by solving the master problem considering only columns used at least once in a solution of a linear relaxation computed while generating columns at the root node. The computation of this upper bound takes only a couple of seconds.

Finally, Table 13 presents the comparison between the results of the PDPS, the PDPTW and the OVRPTW. The savings that can be achieved by the OVRPTW compared to the PDPS solution can reach up to 22.68%. The savings that can be achieved from the PDPTW lower bounds compared to the PDPS solution are, of course, smaller as the PDPTW is an intermediate solution between the PDPS and the OVRPTW solutions. They range from limited (0.49%) to interesting (9.59%). Unfortunately, the instances with 81 and 109 requests have a large gap between the lower and upper bounds for the PDPTW solution. Therefore, the real benefits of transfers cannot be measured with much accuracy. The reported savings of 3.45% and 5.20% are likely to be underestimated. In any case, these tests show that the pooling of the transport between the centers, and particularly the use of transfer, can bring significant savings. However, these savings vary a great deal from one instance to another.

6. Conclusion

Consolidation of requests at transfer points is likely to improve the efficiency of complex door-to-door transportation systems. In this paper, we proposed a graph model for transfer points, an arc-based and two path-based models for the Pickup and Delivery Problem with Shuttle routes (PDPS). We described a branch-and-cut-and-price method to solve the PDPS. The branch-and-price algorithm raises two sub-problems: an elementary shortest path problem with resource constraints solved by a labeling algorithm and a simple sorting problem. The linear relaxation of original models is reinforced by considering three types of valid inequality, which considerably speeds up the calculation.

We show that the path-based models can lead to optimal solutions of a large variety of instances. Within one hour, all generated instances with 25 requests are solved to optimality as well as almost all instances with 50 requests and half of the instances with 75 requests. One real-world instances with 87 requests could also be solved to optimality. Even when the instances cannot be solved within a time limit of one hour, the optimality gap observed is generally less than 1%. Hence, this study is a step toward the efficient solving of the Pickup and Delivery Problems with Transfers (PDPT).

We also assessed the cost savings brought by transfers compared to two approaches: solving independent VRPTWs with open routes (OVRPTW) for each delivery location or solving one classical

Pickup and Delivery Problem with Time Windows (PDPTW) without considering any transfer opportunity. We show large savings variations from one instance to another. It can reach 9% when compared with the PDPTW cost and 22% if compared with the OVRPTW cost.

An important criterion for measuring the quality of service to users is the ride time, *i.e.* the time spent by each passenger in the vehicles. Associating each request with a maximum ride time transforms the problem into a Dial-a-Ride-Problem (DARP). However, it also considerably complicates route scheduling. Considering a heterogeneous fleet is also an interesting avenue to make the problem more realistic. More generally, identifying the characteristics of instances that may greatly benefit from transfer points and locating these transfer points are challenging research issues.

Acknowledgments

The authors thank the doctoral school ED STIM for funding the research stay of Renaud Masson at the DTU. They are grateful to the Conseil Général de Loire-Atlantique and the ADAPEI 44 for their authorization to use their data. The work of Stefan Ropke was sponsored by the Danish Agency for Science, Technology and Innovation (Project “Intelligent Freight Transport Systems”).

Appendix A. Appendix: Arcs removed from A to break symmetries and reinforce the linear relaxation of the problem

The following sets of arcs are empty, in order to break some symmetries and reinforce the linear relaxation of the problem:

- $\delta^+(i) \cap \delta^-(o') = \emptyset, \forall i \in P \cup T^+$.
- $\delta^+(o) \cap \delta^-(i) = \emptyset, \forall i \in D \cup T^-$.
- $\delta^+(i) \cap \delta^-(j) = \emptyset, \forall i \in D, j \in T^-$.
- $\delta^+(d_i) \cap \delta^-(d_j) = \emptyset, \forall i \in R, j \in R | j < i$.
- $\delta^+(t_{r,\tau}^-) \cap \delta^-(t_{r',\tau}^-) = \emptyset, \forall \tau \in T, r \in R, r' \in R | r' < r$.
- $\delta^+(t_{r,\tau}^+) \cap \delta^-(t_{r',\tau}^+) = \emptyset, \forall \tau \in T, r \in R, r' \in R | r' < r$.

No arc between a pickup node or an outbound transfer node and the ending depot should be used. All requests must be unloaded at a delivery point before the vehicle goes back to the depot. The service duration at the delivery point is only accounted for once when multiple requests are unloaded simultaneously. Some equivalent solutions exist. In order to avoid these equivalent solutions, we assume that (i) all transfer operations are performed before delivery operations and (ii) requests are forced to be unloaded in decreasing order of their index at these points. These constraints only help to prevent symmetrical solutions and limit the risk of encountering sub-tours in solutions of the linear relaxation of the problem.

References

- Baldacci, R., Bartolini, E., & Mingozzi, A. (2011). An exact algorithm for the pickup and delivery problem with time windows. *Operations Research*, 59(2), 414–426.
- Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey. *TOP*, 15, 1–31.
- Cordeau, J.-F., Laporte, G., & Ropke, S. (2008). Recent models and algorithms for one-to-one pickup and delivery problems. In B. Golden, R. Raghavan, & E. Wasil (Eds.), *The vehicle routing problem: Latest advances and new challenges* (pp. 327–357). Springer.
- Cortés, C. E., Matamala, M., & Contardo, C. (2010). The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3), 711–724.
- Desaulniers, G., Desrosiers, J., & Sporendonk, S. (2011). Cutting planes for branch-and-price algorithms. *Networks*, 58(4), 301–310.
- Feillet, D., Dejax, P., Gendreau, M., & Guéguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44, 216–229.

- Fugenschuh, A. (2009). Solving a school bus scheduling problem with integer programming. *European Journal of Operational Research*, 193(3), 867–884.
- Gørtz, I. L., Nagarajan, V., & Ravi, R. (2009). Minimum makespan multi-vehicle dial-a-ride. In *Algorithms – ESA 2009*. In A. Fiat & P. Sanders (Eds.), *Lecture notes in computer science* (Vol. 5757, pp. 540–552). Berlin Heidelberg: Springer.
- Irnich, S., & Desaulniers, G. (2005). Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, & M. Solomon (Eds.), *Column generation* (pp. 33–65). Springer.
- Jepsen, M., Petersen, B., Spoorendonk, S., & Pisinger, D. (2008). Subset-row inequalities applied to the vehicle routing problem with time windows. *Operations Research*, 56(2), 497–511.
- Kerivin, H. L. M., Lacroix, M., Mahjoub, A. R., & Quilliot, A. (2008). The splittable pickup and delivery problem with reloads. *European Journal of Industrial Engineering*, 2(2), 112–133.
- Kohl, N., Desrosiers, J., Madsen, O. B. G., Solomon, M. M., & Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33, 101–116.
- Laporte, G., Desrochers, M., & Nobert, Y. (1984). Two exact algorithms for the distance constrained vehicle routing problem. *Networks*, 14, 161–172.
- Lehuédé, F., Pavageau, C., & Péton, O. (2009). A tabu search based heuristic for the transportation of mentally disabled people. In *ORAHs 2009*. Leuven Belgique.
- Lin, C. K. Y. (2008). A cooperative strategy for a vehicle routing problem with pickup and delivery time windows. *Computers & Industrial Engineering*, 55(4), 766–782.
- Lysgaard, J. (2003). CVRPSEP: a package of separation routines for the capacitated vehicle routing problem. Working paper 03-04, Department of Management Science and Logistics, Faculty of Business Administration, The Aarhus School of Business.
- Lysgaard, J., Letchford, A. N., & Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100, 423–445.
- Masson, R., Lehuédé, F., & Péton, O. (2013a). An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47, 344–355.
- Masson, R., Lehuédé, F., & Péton, O. (in press). The dial-a-ride problem with transfers. *Computers & Operations Research*.
- Mitrović-Minić, S., & Laporte, G. (2006). The pickup and delivery problem with time windows and transshipment. *INFOR*, 44(3), 217–228.
- Mues, C., & Pickl, S. (2005). Transshipment and time windows in vehicle routing. In *ISPAN'05: Proceedings of the 8th international symposium on parallel architectures, algorithms and networks* (pp. 113–119). Washington, DC, USA: IEEE Computer Society.
- Naddef, D., & Rinaldi, G. (2002). *Branch-and-cut algorithms for the capacitated VRP*. *SIAM monographs on discrete mathematics and applications* (pp. 53–84).
- Nakao, Y., & Nagamochi, H. (2010). Worst case analysis for pickup and delivery problems with transfer. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E91-A(9), 2328–2334.
- Oertel, P. (2000). Routing with reloads. Ph.D. thesis, University of Cologne, Germany.
- Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2006). A survey on pickup and delivery problems. Part 2: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*.
- Petersen, H. L., & Ropke, S. (2011). The pickup and delivery problem with cross-docking opportunity. In J. W. Bse, H. Hu, C. Jahn, X. Shi, R. Stahlbock, & S. Voß (Eds.), *Computational logistics. Lecture notes in computer science* (Vol. 6971, pp. 101–113). Berlin Heidelberg: Springer.
- Qu, Y., & Bard, J. F. (2012). A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research*, 39(10), 2439–2456.
- Ropke, S., & Cordeau, J.-F. (2009). Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3), 267–286.
- Russell, R. A., Morrel, R. B., & Haddox, B. (1986). Routing special-education school buses. *Interfaces*, 16(5), 56–64.
- Shang, J. S., & Cuff, C. K. (1996). Multicriteria pickup and delivery problem with transfer opportunity. *Computers & Industrial Engineering*, 30(4), 631–645.
- Thangiah, S. R., Fergany, A., & Awam, S. (2007). Real-time split-delivery pickup and delivery time window problems with transfers. *Central European Journal of Operations Research*, 15, 329–349.
- Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. *SIAM monographs on discrete mathematics and applications*. Philadelphia: SIAM.