UNIVERSITÉ DE TECHNOLOGIE DE COMPIEGNE

Habilitation à diriger des recherches de l'Université de Technologie de Compiègne

Présentée et soutenue publiquement par

Caroline PRODHON

Modèles et méthodes d'optimisation pour des problèmes difficiles de localisation et tournées de véhicules

Année : 2012

Jury

Rapporteurs :	Dominique FEILLET	*Professeur, CMP Georges Charpak, Ecole des Mines de Saint-Etienne*
	Aziz MOUKRIM	*Professeur, HeuDiaSyC, Université de Technologie de Compiègne*
	Bernard PENZ	*Professeur, G-SCOP, Institut Polytechnique de Grenoble*
	Marc SEVAUX	*Professeur, Lab-STICC, Université de Bretagne-Sud*
Examinateurs :	Nacima LABADIE	*MdC (HDR), ICD, Université de Technologie de Troyes*
	Philippe LACOMME	*MdC (HDR), LIMOS, Université de Clermont-Ferrand*
	Christian PRINS	*Professeur, ICD, Université de Technologie de Troyes*
	Eric TAILLARD	*Professeur, MIS, Haute Ecole d'Ingénierie et de Gestion du canton de Vaud*

Résumé

Ce mémoire en vue de l'habilitation à diriger des recherches résume mes activités durant mes dix premières années de recherche, depuis mon stage de DEA jusqu'à ce jour. Après une introduction, une première partie est consacrée à mon parcours avec mes différentes activités pédagogiques, administratives et scientifiques. Une autre partie du mémoire se concentre sur ma recherche, avant de terminer par une conclusion générale et des perspectives.

La thématique essentielle de mes activités porte sur la modélisation et le développement de méthodes d'optimisation pour des problèmes difficiles de localisation et tournées de véhicules, avec contraintes de capacité. Il s'agit d'optimiser la logistique du transport pour des problèmes pouvant impliquer différents niveaux de décision (stratégique avec des décisions de localisation de dépôts, tactique concernant le choix de véhicules ou l'affectation de clients, et opérationnelle pour l'élaboration de tournées de véhicules et/ou la gestion de stock). Ces niveaux sont souvent inter-dépendants, mais pour des raisons de simplification, ils sont généralement traités séparément. Or, l'optimisation globale permet des gains significatifs sur les coûts totaux. La motivation des travaux développés ici est d'aborder des problèmes difficiles et dans des versions encore peu étudiées dans la littérature, comportant des ressources limitées aussi bien en terme de capacités (au niveau des dépôts à ouvrir, ou des véhicules réalisant les tournées, ou bien encore des zones d'entreposage chez les clients) qu'en terme de quantité (nombre de véhicules limité par exemple). Le cas multi-objectif est aussi envisagé. De plus, des problèmes de taille réaliste sont visés.

L'étude se fait par l'élaboration d'algorithmes de résolution prenant en considération l'intégralité du problème, sans décomposition hiérarchique en phases. Pour cela, les méthodes peuvent être des approches exactes basées sur la proposition de nouveaux modèles mathématiques, mais également avec des méthodes de type heuristique et hybride. Une implémentation est également proposée avec une coopération entre procédures exécutées en parallèle. Tous les algorithmes développés sont testés et validés sur des jeux d'essais nouveaux ou provenant de la littérature.

Mots clés : Optimisation combinatoire, Recherche opérationnelle, Logistique, Problèmes de tournées de véhicules, Problèmes de localisation.

Remerciements

La rédaction de ce manuscrit en vue de l'HDR m'a permis de faire un bilan de mon activité de recherche menée au cours des dernières années. Mais, cette activité est indissociable des gens rencontrés. Cette partie est donc très importante puisqu'elle est dédiée à toutes les personnes qui m'ont permis d'en arriver où je suis.

Je tiens à exprimer ma reconnaissance aux membres du jury qui a évalué ce travail. Tout d'abord, je remercie les rapporteurs pour avoir pris le temps de lire ce mémoire dans les moindres détails et les examinateurs pour leur participation à ma soutenance.

Plus particulièrement, ma gratitude va à Marc Sevaux, que j'ai toujours plaisir à retrouver, pour sa confiance et son estime. Je remercie également Dominique Feillet pour sa présence malgré ses contraintes dans la période de soutenance, Aziz Moukrim pour ses encouragements sa sympathie et Nacima labadie pour ses encouragements, ses conseils et les agréables moments partagés dans le laboratoire. Par cette HDR, j'ai eu l'honneur de faire la connaissance d'Eric Taillard, président du jury, avec qui j'ai pu avoir des échanges constructifs, et de Bernard Penz que j'ai eu grand plaisir de rencontrer.

Je tiens à remercier singulièrement Philippe Lacomme, qui a jadis été mon conseiller en tant qu'étudiante ingénieur, et que j'ai eu le plaisir de retrouver quelques années plus tard en tant que collègue. Les heures partagées de travail et discussions productives sont un réel plaisir et ont découlé sur une amitié sincère. J'apporte toute ma reconnaissance à Christian Prins pour son aide, sa rigueur et ses précieuses recommandations qui m'ont aidée à prendre du recul. Il m'a placée dans des conditions de travail idéales depuis ma thèse.

Merci à toutes les personnes avec qui j'ai travaillé avec un immense plaisir, Jose-Manuel Belenguer, Enrique Benavent, Christophe Duhamel, Demetrio Macias, Andres Medaglia, Angel Ruiz, Patrick Soriano, Nubia Velasco, Roberto Wolfler-Calvo et tous ceux avec qui j'ai pu être en contact dans les laboratoires.

Un très grand merci à tous les doctorants que j'ai pu co-encadrer ou que j'encadre encore actuellement : Juan-Guillermo Villegas, Viet-Phuong Nguyen et William Guerrero. Par tous nos échanges fructueux, ce travail est aussi le leur.

Ces remerciements ne seraient pas complets sans amitié dédiée à l'ensemble de l'équipe "systèmes logistiques" du LOSI, particulièrement Murat Afsar, et les secrétaires Marie-José et Véronique.

Je désire terminer, en remerciant ma famille pour leur amour et l'équilibre qu'ils m'apportent, ainsi que leur soutien et confiance, et tous ceux que j'aurai pu oublier.

Caroline PRODHON.

Table des matières

Partie 1

Introduction Générale

.

1.1 Contexte

Après un stage de DEA portant sur le paramétrage de la méthode MRP et ayant débouché sur des présentations en conférences et des publications, mes activités de recherche se sont inscrites dans le domaine de la logistique du transport et des tournées de véhicules.

Ma thèse de doctorat s'est déroulée à l'institut Charles Delaunay, dans le Laboratoire d'Optimisation des Systèmes Industriels (LOSI) de l'Université de Technologie de Troyes (UTT). J'ai ensuite été recrutée dans cette équipe et mes travaux s'inscrivent dans l'axe 1 du LOSI, "Systèmes Logistiques".

L'intérêt pour les problèmes de logistique du transport est grandissant dans la vie actuelle puisqu'ils peuvent s'appliquer à de nombreuses activités comme le transfert de flux physiques avec la distribution de courrier, la livraison de colis, la collecte de produits, mais aussi à de plus vastes domaines comme l'agencement de lignes de transports en commun, le traitement du réseau routier ou le ramassage des ordures.

De plus, avec la mondialisation des marchés et les exigences des clients autant en qualité qu'en délais, les coûts logistiques dans les frais des entreprises prennent une part de plus en plus importante. C'est pourquoi ces dernières ont tout intérêt à s'intégrer dans des réseaux d'approvisionnement et de distribution pour tenter de réduire au maximum les coûts engendrés. Il ne faut pas non plus négliger un autre aspect prenant une dimension d'un intérêt grandissant : la prise de conscience environnementale qui incite à une réduction des émissions des gaz à effet de serre, dont le transport est en partie responsable.

C'est pour toutes ces raisons que des efforts sont nécessaires afin de développer des systèmes de transport flexibles et efficaces répondant aux désirs actuels relatifs à l'économie et à notre qualité de vie.

1.2 Problématiques

Parmi le vaste domaine de la logistique du transport, les thématiques abordées concernent essentiellement les problèmes de localisation et tournées de véhicules. Plus précisément, il s'agit de traiter des variantes, impliquant des questions réalistes difficiles encore peu étudiées dans la littérature, et qui concernent une gestion de ressources limitées aussi bien en terme de capacités (au niveau des dépôts approvisionnant des clients, des véhicules réalisant les tournées, ou bien encore des possibilités d'entreposage chez les clients) qu'en terme de quantité (comme une restriction du nombre de véhicules disponibles).

D'autres problématiques d'optimisation sont également traitées mais de manière plus marginale, et comprenant parfois le cas multi-critères.

1.3 Méthodologie

Les décisions à prendre pour des problèmes de transport sont généralement modélisées par des variables binaires ou entières (ouverture ou non d'un dépôt, choix des arcs pour construire les tournées, ...) et l'objectif est d'optimiser les coûts. Elles soulèvent donc des problèmes d'Optimisation Combinatoire (*Combinatorial Optimization - CO*).

La résolution de tels problèmes peut sembler facile, mais en pratique, la quête de l'optimum s'avère souvent très coûteuse en temps de calcul. Il est alors intéressant d'avoir recours aux algorithmes de la Recherche Opérationnelle (*RO*), telles que les heuristiques ou les méthodes arborescentes par exemple.

Les approches proposées ici sont des techniques de résolution dédiées aux problématiques traitées en gardant une considération globale du problème, sans décomposition hiérarchique en niveaux de décision. Trois approches sont développées : des méthodes exactes basées sur de nouveaux modèles mathématiques, des heuristiques/métaheuristiques et des algorithmes hybrides. Dans ce dernier cas, une implémentation est également proposée avec une coopération entre procédures exécutées en parallèle.

1.4 Structure de ce mémoire

Après cette introduction permettant de situer brièvement les travaux, ce mémoire se poursuit par une partie consacrée au curriculum vitæ. Outre les informations générales me concernant, cette section présente un récapitulatif de mon parcours, le résumé de mes différentes activités liées à l'enseignement, celles en relation avec la recherche, mes activités scientifiques et la liste de mes publications.

La partie suivante du mémoire se concentre sur ma recherche. L'ensemble des problématiques étudiées sont détaillées tout en faisant le lien des unes par rapport aux autres. Les approches de résolutions sont aussi exposées, en faisant ressortir les versions me semblant les plus prometteuses, et quelques exemples de résultats sont présentés.

Pour terminer, ce mémoire comporte une dernière partie dédiée aux conclusions et perspectives de travaux, et en annexe quatre articles de revue sélectionnés parmi la liste de mes publications.

Partie 2

Curriculum Vitæ

.

2.1 Informations Générales

2.1.1 Etat Civil

Nom	: PRODHON
Prénom	: Caroline
Date de naissance	: 25 août 1979
Lieu de naissance	: Chaumont (52)
Nationalité	: Française
Situation familiale	: Pacsée, 1 enfant
Adresse personnelle	: 11 rue Jeannette 10000 Troyes - France
Tel personnel	: ++033 310 95 42 14

2.1.2 Situation actuelle

Fonction	: Maître de Conférences depuis le 01/09/2007
Section CNU	: 27^{e} (informatique)
Etablissement de rattachement	: Université de Technologie de Troyes (UTT) Institut Charles Delaunay (ICD) UMR CNRS 6279 - Sciences et Technologies pour la Maîtrise des Risques (STMR) Laboratoire d'Optimisation des Systèmes Industriels (LOSI) Département des Systèmes Industriels (SI)
Adresse professionnelle	: 12 rue Marie Curie BP 2060, 10010 Troyes Cedex - France
Tel professionnel	: ++033 325 25 71 84 47
E-mail	: caroline.prodhon@utt.fr
Page personnelle	: http://prodhonc.free.fr/
Domaine de Recherche	: Recherche Opérationnelle - Optimisation Combinatoire Logistique du Transport - Gestion de Production
Prime d'Excellence Scientifique	: Attribution en 2011

2.2 Parcours et Formation

Depuis 2007 : **Maître de conférences**, section 27 - UTT

2006 - 2007 : **ATER** section 27/61 - UTT

2003 - 2006 : **Thèse de doctorat** en Optimisation des Systèmes
Soutenue le 16/10/2006 à l'UTT - **Mention Très Honorable**
(l'UTT ne délivre plus de félicitations)
Sujet : Le problème de Localisation-Routage

Encadrants :	Christian PRINS, Professeur *Institut Charles Delaunay*, UTT Roberto WOLFLER-CALVO, Professeur *LIPN*, Université Paris 13
Rapporteurs :	Michel GENDREAU, Professeur *CIRRELT,* Université de Montréal, Canada Frédéric SEMET, Professeur *LAGIS*, École Centrale de Lille
Examinateurs :	Alexandre DOLGUI, Professeur *G2I*, École des Mines de Saint-Étienne Aristide MINGOZZI, Professeur *PSDC*, Université de Bologne, Italie Marc SEVAUX, Professeur *Lab-STICC*, Université de Bretagne-Sud

\+ **Monitorat** en Mathématiques - UTT

Mai 2006 (1 semaine) : École printanière sur les **tournées de véhicules**
(séminaires avancés pour doctorants) - HEC Montréal, Canada

Octobre 2003 (1 semaine) : École d'automne de **recherche opérationnelle**
(séminaires avancés pour doctorants) - Polytech'Tours, France

2002 - 2003 : **DEA** en Optimisation et Sûreté des Systèmes
Sujet : Planification des approvisionnements en milieu incertain : Paramétrage du système MRP
Encadrants : Alexandre DOLGUI et Aly OULD-LOULY.
\+ formation **Ingénieur en Génie des Systèmes Industriels**
(option Gestion de Production) - UTT

2.3 Enseignement

En tant qu'enseignant-chercheur, mon activité se consacre en partie à l'enseignement supérieur. Outre la transmission des connaissances, je participe à la direction et au conseil des études par mes responsabilités de formation.

2.3.1 Contexte et organisation de l'enseignement

L'Université de Technologie de Troyes (UTT) est une école d'ingénieur qui recrute ses étudiants sur dossier et entretien individuel. Deux principaux niveaux d'accès sont possibles :

- niveau Bac (S) avec une formation sur 5 ans : 2 années préparatoires, dites de *Tronc commun*, puis 3 ans de spécialisation dites de *Branche*.
- à Bac +2 avec une formation sur 3 ans de *Branche*.

La dernière année est une spécialisation, appelée *Filière*.

L'année universitaire à l'UTT est divisée en deux semestres de 17 semaines, séparés par un inter-semestre de 4 semaines. L'enseignement est dispensé par unités de valeur (UV) qui chacune correspond à une quantité de travail relative au nombre de crédits ECTS (European Credit Transfer System) qu'elle attribue. Chaque étudiant ingénieur en choisis en moyenne 6 par semestre répartis entre différentes catégories (CS - connaissances scientifiques, TM - techniques et méthodes, EC - expression et communication, ME - management de l'entreprise, et CT - culture et technologie) afin de compléter un profil de formation. Durant leur études, deux semestres complets sont réservés à des stages en entreprise (TN09 pour le stage professionnel et TN10 pour le stage de fin d'études).

Outre, le diplôme d'ingénieur avec 6 spécialisations, l'UTT décerne deux licences professionnelles, un Master en Sciences, Technologies et Santé (avec 3 mentions et 9 spécialités), ainsi qu'un Doctorat Sciences des systèmes technologiques et organisationnels (avec cinq champs disciplinaires).

Mon intervention se situe dans des cours de type CS/TM essentiellement pour des élèves ingénieurs en branche systèmes industriels (SI), ainsi que dans le Master SMILES (Sport, Management et Ingénierie, Logistique Evénementielle et Sécurité).

J'ai aussi été sollicitée pour un cours de planification dans une licence professionnelle délivrée par l'IUT de Troyes : Gestion et Organisation Logistique (GOL), pour une charge de 35h de cours par an.

2.3.2 Disciplines enseignées

GP06 : Organisation et gestion de la production

- niveau Bac+3
- volume horaire enseigné : 32h de TD par groupe et par semestre
  + 32h de cours par semestre
- nombre d'étudiants par groupe de TD : 35
- nombre d'étudiants par cours : de 180 à 220
- activités reliées : préparation des cours dispensés en amphithéâtre, élaboration d'un poly de cours, surveillance des examens, participation à l'élaboration des sujets d'examens, correction d'examens, participation au jury d'attribution du module.

Programme enseigné : Contexte technico-économique et typologie de production - connaissances spécifiques : codification, classification, prévisions, implantations, planification, ordonnancement, lancement, suivi, analyse de la valeur, coût de revient - méthodes de gestion et domaines d'utilisation : gestion des stocks, MRP, Kanban, MPM, OPT - organisation de la gestion de production et performance de l'entreprise : réactivité, flexibilité, tension des flux, réduction des gaspillages.

LO01 : Bases de l'informatique

- niveau Bac+3
- volume horaire enseigné : 32h de TD par groupe et par semestre
- nombre d'étudiants par groupe de TD : 25
- activités reliées : surveillance d'examens, élaboration et correction de contrôle continu, participation au jury d'attribution du module.

Programme enseigné : Introduction - contexte de développement d'une application - automate et langage - place du langage de programmation et lien avec un système automatique - architecture d'un ordinateur - algorithmique - concepts et règles de base pour la conception d'algorithmes - introduction au langage C - les fondements - structures de données - tableaux, fichiers, articles... - programmation avancée - structures de données dynamiques et récursivité.

NF04 : Algorithmique

- niveau Bac+1/2
- volume horaire enseigné : 32h de TD par groupe et par semestre
- nombre d'étudiants par groupe de TD : 25

- activités reliées : surveillance d'examens, élaboration et correction de contrôle continu, participation au jury d'attribution du module.

Programme enseigné : démarche algorithmique de résolution des problèmes - éléments de complexité algorithmique et analyse d'algorithmes - structures de données simples (tableau, chaînes, articles...) - exemples d'algorithmes de résolutions de problèmes classiques (tris, recherche d'éléments, calculs numériques élémentaires...) - introduction à l'architecture de l'ordinateur - représentation interne des données et des instructions.

SY18 : Outils de modélisation et d'évaluation des performances

- niveau Bac+2
- volume horaire enseigné : 32h de TD par groupe et par semestre
- nombre d'étudiants par groupe de TD : 30

Programme enseigné : modèles et algorithmes basés sur les graphes - réseaux de Petri - chaînes et processus de Markov - files d'attente

IF08 : Management de projets informatiques

- niveau Bac+3
- volume horaire enseigné : 2h de TD par groupe et par semestre
  +2h de cours par semestre
- nombre d'étudiants par groupe de TD : 25
- nombre d'étudiants par cours : 100
- activités reliées : préparation des cours dispensés

Programme enseigné : les principes et techniques du management de projet - planification, gestion des ressources.

CS03 : Conduite de projets

- niveau Bac+3
- volume horaire enseigné : 2h de TD par groupe et par semestre
  + 2h de cours par semestre
- nombre d'étudiants par groupe de TD : 35
- nombre d'étudiants par cours : 150
- activités reliées : préparation des cours dispensés

Programme enseigné : les principes et techniques du management de projet - méthode des potentiels, diagramme de Gantt, gestion des ressources.

IS08 : Modélisation de la logistique événementielle

- niveau Bac+5
- volume horaire enseigné : 10h de TD par semestre
  + 10h de cours par semestre
- nombre d'étudiants : 25
- activités reliées : préparation des cours dispensés, surveillance des examens, participation à l'élaboration des sujets d'examens, correction d'examens, participation au jury d'attribution du module.

Programme enseigné : outils de gestion de projet et d'aide à la décision - estimation des besoins et des ressources nécessaires - ordonnancement - conduite des actions / management.

MT22 : Fonctions de plusieurs variables et applications

- niveau Bac+1/2
- volume horaire enseigné : 32h de TD par groupe et par semestre
- nombre d'étudiants par groupe de TD : 25
- activités reliées : surveillance des examens, participation à l'élaboration des sujets d'examens, correction d'examens, participation au jury d'attribution du module.

Programme enseigné : Généralisation des notions de limite et de continuité aux fonctions de plusieurs variables, différentiabilité - analyse vectorielle : les opérateurs utilisés en physique et mathématiques pour l'ingénieur - courbes et surfaces : étude locale, paramétrisation - intégrales doubles, triples, curvilignes, intégrales de surface - théorèmes intégraux de l'analyse vectorielle et applications à l'électromagnétisme.

NF14 : Gestion des systèmes industriels assistée par ordinateur

- niveau Bac+4/5
- volume horaire enseigné : 8h de TP par groupe et par semestre
- nombre d'étudiants par groupe de TP : 15
- activités reliées : participation au jury d'attribution du module.

Programme enseigné : utilisation des systèmes assistés par ordinateur en entreprise - gestion de production (GPAO) : données techniques, planification à moyen terme, calcul des besoins, gestion des approvisionnements, ordonnancement et lancement

SY15 : Simulation des systèmes industriels

- niveau Bac+3/4
- volume horaire enseigné : 2h de cours par semestre
- nombre d'étudiants : 70

Programme enseigné : introduction aux méthodes d'optimisation utilisant la simulation.

TN : Suivis de stages étudiants

Chaque enseignant-chercheur à l'UTT doit suivre des étudiants en stage de 6 mois en entreprise. Il s'agit aussi bien de stage technique que de projet de fin d'études. Généralement, un total de 4 à 5 stagiaires est attribué chaque semestre. Le but de ce suivi est de s'assurer du bon déroulement du stage, si possible en s'y rendant sur place, et d'évaluer la qualité du rapport ainsi que de la soutenance. L'évaluation orale se faisant en présence de 2 enseignants-chercheurs, il est de convenance d'assister aux soutenances des étudiants de l'enseignant jugeant les stagiaires dont nous avons le suivi.

TX : Suivis de projets étudiants

Il s'agit d'un travail de définition, réalisation et mise en oeuvre d'un ou plusieurs dispositifs techniques sous la direction d'un enseignant.

GOL : Planification

Ce cours intervient à l'IUT.

- niveau Bac+3
- volume horaire enseigné : 35h par semestre
- nombre d'étudiants : 25
- activités reliées : préparation des cours et TD, participation à l'élaboration des sujets d'examens, correction d'examens.

Programme enseigné : Contexte technico-économique et typologie de production - connaissances spécifiques : codification, classification, prévisions, implantations, planification, ordonnancement, MRP, Kanban, OPT, réduction des gaspillages.

2.3.3 Activités d'enseignement

Outre les enseignements effectués, récapitulés dans la section ci-dessous, et les implications qui y sont liées (en particulier pour les UV dont nous sommes responsables), d'autres activités liées à la formation m'ont été confiées, comme cela est décrit dans la suite.

2.3.3.1 Récapitulatif des enseignements dispensés

Le tableau 2.1 donne un résumé par année des interventions que j'ai réalisées dans les matières détaillées dans la section 2.3.2. Depuis ma nomination en tant que maître de conférences en 2007, j'effectue un service supérieur au taux obligatoire requis qui est de 192h équivalent TD. Ceci est dû à la sous-dotation de l'UTT en personnel enseignant.

A noter que pour l'année universitaire 2010/2011, j'ai bénéficié d'une décharge d'enseignement due à mon congé de maternité.

Responsable d'unités de valeurs :

GP06 : Organisation et gestion de la production *(220 étudiants),*
IS08 : Modélisation de la logistique événementielle *(25 étudiants)*

2.3.3.2 Responsabilités de formation

Rôle de conseiller

Comme tous les enseignants de l'UTT, je suis conseiller de plusieurs étudiants de l'UTT (généralement entre 15 et 20 par an).

Le rôle est d'apporter un soutien et des conseils notamment aux étudiants confrontés à des difficultés ou à des interrogations. En cas de difficulté de réussite aux examens, je participe au 2ème jury de suivi pour apporter un éclairage du parcours de l'étudiant.

Responsabilité de cours

Chaque UV est gérée par un enseignant-chercheur qui a des responsabilités pédagogiques mais aussi organisationnelles (conception et mise à jour du contenu, constitution et coordination de l'équipe enseignante, organisation des examens, ...).

En tant que responsable des UV GP06 (depuis 2008 - effectif de 180 à 220 étudiants par semestre) et IS08 (depuis 2007 - effectif de 20 à 30 étudiants par semestre), j'ai donc en charge l'organisation de ces modules.

Années	UV	Nombre d'heures (équivalent TD)	Total (équivalent TD)	Statut
2003/2004	MT22	64	64 (64 dues)	Moniteur
2004/2005	MT22 NF04	32 32	64 (64 dues)	Moniteur
2005/2006	MT22 GP06	32 32	64 (64 dues)	Moniteur
2006/2007	GP06 LO01 TN	43 34 20	97 (96 dues)	1/2 ATER
2007/2008	GP06 LO01 SY18 IS08 NF14 CS03 TN	84 68 34 20 8 3 35	252 (192 dues)	MdC
2008/2009	GP06 LO01 IS08 NF14 CS03 IF08 TN	103 36 28 8 8 8 69	260 (192 dues)	MdC
2009/2010	GP06 LO01 IS08 NF14 CS03 IF08 SY15 TX TN	80 32 28 4 9 8 3 5 67	236 (192 dues)	MdC
2010/2011	GP06 LO01 IS08 IF08 TN	71 34 28 8 20	161 (144 dues)	MdC Congé de maternité
2011/2012	GP06 LO01 IS08 IF08 GOL TN TX	103 34 28 8 52 20 5	250 (192 dues)	MdC

Table 2.1: Récapitulatif des heures enseignées par année

Responsabilité de filière de la formation ingénieur

Je suis responsable **de la filière Logistique Externe et Transport** (LET) de la formation ingénieur Systèmes Industriels (SI) de l'UTT depuis janvier 2010. Cette filière, la plus importante de l'UTT, est une des 3 spécialités de fin de cycle de la formation ingénieur en Systèmes Industriels (SI). Chaque année, la filière LET compte environs 150 étudiants, ce qui représente plus de la moitié des étudiants de la formation SI.

Mon rôle est d'être le garant des enseignements de la filière en validant le cursus des étudiants, les enseignements de filières suivis à l'étrangers, les sujets de stage de la filière, et en participant au suivi pédagogique, jury de suivi et au jury de diplôme. J'ai pour mission également de faire évoluer l'offre de formation de la filière en accord avec le responsable de formation.

Jurys

Je participe régulièrement à plusieurs des jury suivants :

- jury d'admission
- jury de validation d'UV
- jury de suivi semestriel
- jury de diplôme

2.4 Activités et animation de la recherche

En dehors des travaux de recherches et des publications (résumées dans la section 2.4.1), cette section détaille les activités administratives et les activités ayant trait à l'animation et au rayonnement de la recherche.

Notons qu'au cours de l'évaluation 2011, la Prime d'Excellence Scientifique m'a été attribuée.

2.4.1 Synthèse des publications

Le principal produit de ma recherche se réalise sous forme de publications scientifiques dans des revues ou des conférences internationales. Avec 15 papiers dans des livres ou revues de renom (répertoriées au Journal Citation Report) sur 7 ans, cela représente une moyenne de plus de 2 articles par an, et presque 3 sur les 3 dernières années.

La répartition par type de publication est la suivante :

- Chapitres de livre : 1 chapitre publié, 1 en cours de révision
- Revues internationales : 14 articles publiés, 1 en cours de révision
- Conférences internationales avec comité de lecture et actes publiés : 14
- Conférences internationales avec comité de lecture sans actes publiés : 18
- Conférences nationales avec comité de lecture et actes publiés : 2
- Conférences nationales avec comité de lecture sans actes publiés : 9

D'après le logiciel Publish or Perish version 3.3 (Harzing), au 28/02/2012, voici le résultat de ma recherche:

<table>
<tr><td>Papers: 33
Citations: 260
Years: 8
Cites/year: 32.50</td><td>Cites/paper: 7.88
Cites/author: 95.83
Papers/author: 13.45
Authors/paper: 3.18</td><td>h-index: 7
g-index: 15
hc-index: 7
hI-index: 2.13
hI,norm: 6</td><td>AWCR: 58.82
AW-index: 7.67
AWCRpA: 21.03
e-index: 12.53
hm-index: 4.95</td></tr>
<tr><td colspan="4">Query date: 2012-02-28 Hirsch a=5.31, m=0.88 Contemporary ac=4.80
Cites/paper 7.88/3.0/0 (mean/median/mode)
Authors/paper 3.18/3.0/3 (mean/median/mode)</td></tr>
<tr><td colspan="2">5 paper(s) with 1 author(s)
3 paper(s) with 2 author(s)
12 paper(s) with 3 author(s)</td><td colspan="2">7 paper(s) with 4 author(s)
6 paper(s) with 5 author(s)</td></tr>
</table>

Table 2.2: Résultat pour "Caroline Prodhon" sur Publish or Perish 3.3

2.4.2 Animation locale

Par animation locale, j'entends ma participation à des activités au sein de mon université, tels que des projets ou des responsabilités administratives.

2.4.2.1 Encadrement de 3^e^ cycle

Co-direction/encadrement de thèses : 3

- **Juan Guillermo Villegas** - Co-encadrement de thèse en cotutelle avec Christian Prins (partie française - UTT), et Andrés Medaglia et Nubia Velasco (partie colombienne - Université des Andes), Soutenue en décembre 2010
  Sujet : Vehicle routing problems with trailers.
- **Viet Phuong Nguyen** - Co-direction de thèse avec Christian Prins, Soutenue en décembre 2011 à l'UTT
  Sujet : Optimisation de problèmes de localisation-routage à 2 niveaux.
- **William Guerrero** - Co-direction de thèse en cotutelle avec Nubia Velasco (partie colombienne - Université des Andes), commencée en Août 2010
  Sujet : Inventory Location Routing Problem.

J'encadre également des étudiants en projet de Master Recherche en laboratoire et en entreprise.

2.4.2.2 Projets de recherche

Je suis porteur des projets notés d'un astérisque.

Projet PIVERT (Picardie Innovations Végétales, Enseignements et Recherches Technologiques).

Durée : 10 ans
Budget : 689 000 euros (juin 2012- juin 2015)

Dans le cadre des investissements d'avenir et des instituts d'excellences, je participe à l'IEED (Institut d'Excellence et d'Energies Décarbonnées) avec le projet PIVERT (Picardie Innovations Végétales, Enseignements et Recherches Technologiques). Ce dernier est spécialisé dans la chimie du végétal, les technologies et l'économie des bioraffineries de troisième génération (valorisation du végétal dans son intégralité) et dans le domaine de la biomasse oléagineuse et forestière. Il réunit trois partenaires académiques (l'UT de Troyes, l'UT de Compiègne et l'Université Picardie Jules Verne) et des acteurs publics et privés (Pôle IAR, Sofiprotéol, Rhodia, Lavalin, Maguin, ...).

Avec la participation de quatre autres collègues du LOSI, le rôle dans ce projet consiste à apporter notre expertise pour la mobilisation des ressources et l'optimisation de la logistique (approvisionnement, transport, stockage, ...) afin d'alimenter les bioraffineries dans un schéma d'économie du carbone.

Projet Google (Matheuristics frameworks - Application to logistics through google maps APIs)

Durée : 1 an (2012/2013)
Budget : 50 000 euros

Le but est de proposer des méthodes d'optimisation avancées de type *Matheuristique*, une hybridation entre approches exactes et métaheuristiques, pour les problèmes de tournées de véhicules. La conception de méthodes s'accompagne du développement d'une interface internet, libre d'accès, s'appuyant sur la librairie Google Maps. Le but est de pouvoir résoudre des problèmes de tournées en ligne.

Projet abondement Carnot-UT (OLIN : Optimisation de la Logistique Inverse)

Durée : 3 ans (février 2010- février 2013)
Budget : 40 000 euros

Le but est de proposer des méthodes d'optimisation pour résoudre des problématiques issues du domaine de la logistique inverse (de la collecte de produits usagés jusqu'à des problèmes combinant planification du transport et de la production dans des processus de désassemblage). J'ai co-encadré dans ce projet le Master Recherche de Nizar Triki avec Nacima Labadie. Le sujet a porté sur un problème de planification du transport avec dépôts multiples et fenêtres de temps et combiné à la gestion de stock.

Action incitative* (Etudes de méthodes d'optimisation appliquées en physique du champ proche)

Durée : 2 ans (2008/2009)
Budget : 3 000 euros

L'objectif principal de ce travail est de mener une activité commune entre le Laboratoire d'Optimisation des Systèmes Industriels (LOSI) et Laboratoire de Nanotechnologie et d'Instrumentation Optique (LNIO) de l'UTT. L'union des compétences de ces groupes permet de réaliser un projet d'étude de méthodes d'optimisation appliquées à des problématiques spécifiques en nanotechnologie.

Le LNIO développe des modèles physiques qu'ils souhaitent utiliser, au delà de l'étude purement électromagnétique, pour réaliser des nano-structures diverses optimisant la performance recherchée. Dans ce but, j'ai apporté, avec l'aide de deux collègues du LOSI, des techniques d'optimisation génériques pour des applications sortant du champ classique des systèmes industriels. Le résultat permet une accélération des procédés tout en minimisant le gaspillage de matériaux.

Projet Exploratoire* (Etude d'approches coopératives pour le problème de localisation-routage périodique)

Durée : 1 an (2008)
Budget : 1 700 euros

Cette étude explore l'utilisation de méthodes d'optimisation dites coopératives pour des problèmes du domaine du transport intégrants divers niveaux de décisions. Le but est de développer des systèmes logistiques flexibles et efficaces. Plus précisément, dans ce projet, l'application porte sur le problème de localisation-routage périodique, qui n'avait jamais encore été étudié. Il correspond à la conception complète d'un réseau de distribution avec la localisation de dépôts, la détermination des jours de visite de chacun des clients de la chaîne logistique selon des fréquences données, et l'élaboration des tournées de véhicules à réaliser.

2.4.2.3 Implication dans les instances universitaires

Membre du **conseil de l'école doctorale** de l'UTT depuis 2008

Ce conseil est réuni deux fois par an pour juger du bilan de l'école doctorale et proposer les grandes orientations concernant en particulier son organisation, son fonctionnement, sa politique d'attribution des allocations et sa politique de suivi des doctorants et des docteurs.

Membre élu du **conseil scientifique** de l'UTT depuis 2010

Le conseil scientifique est consulté sur les orientations des politiques de recherche, de documentation scientifique et technique, ainsi que sur la répartition des crédits de recherche (avis sur les programmes et contrats de recherche). Il assure également la liaison entre l'enseignement et la recherche (consultation sur les programmes de formation, sur la qualification à donner aux emplois d'enseignants-chercheurs et de chercheurs vacants ou demandés, sur les projets liés aux diplômes et sur le contrat d'établissement).

2.4.2.4 Organisation de séminaires

Responsable **des séminaires** au sein du LOSI de Novembre 2008 à Janvier 2010.

2.4.3 Rayonnement national et international

Au delà des publications dans des revues internationales à comité de lecture et la participation à des projets nationaux, mon engagement scientifique et ma reconnaissance se mesurent par diverses contributions.

2.4.3.1 Evaluation de la recherche

Rapporteur de revues internationales :

- Computers & Operations Research,
- European Journal of Operational Research,
- Transportation Science,
- Annals of Operations Research,
- Journal of Operational Research Society,
- Journal of Heuristics,
- Journal of Intelligent Manufacturing,
- International Transactions in OR.

Membre de comités scientifiques de conférences internationales :

- Hybrid Metaheuristics - HM,
- Metaheuristic International Conference - MIC,
- International Conference on Industrial Engineering and Systems Management - IESM.

Expertises de dossiers scientifiques :

- programme "Établissement de nouveaux chercheurs" du Fonds de recherche du Québec.

Membre de comités de sélection pour des postes de Maître de Conférence :

- poste en section 27 pour l'INPG.

2.4.3.2 Collaborations/Séjours de recherche à l'étranger

Collaborations de recherche

Les collaborations de recherche suivantes ont débouché sur des travaux présentés en conférences, mais surtout sur des articles de revue tels que mentionnés ci-dessous.

- *LIMOS - Clermont-Ferrand* avec P. Lacomme et C. Duhamel

C. Duhamel, C. Gouinaud, P. Lacomme, C. Prodhon. A multi-thread GRASPx-ELS for the heterogeneous capacitated vehicle routing problem, dans *Hybrid Metaheuristics*, Springer, en révision.

C. Duhamel, P. Lacomme, C. Prodhon. A hybrid Evolutionary Local Search with depth first search split procedures for the heterogeneous vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 25(2), pp. 345-358, 2012.

C. Duhamel, P. Lacomme, C. Prodhon. Efficient frameworks for greedy split and new depth first search split procedures for routing problems. *Computers & Operations Research*, 38, pp. 723-739, 2011.

C. Duhamel, P. Lacomme, C. Prins, C. Prodhon. A GRASPxELS Approach for the capacitated Location-Routing Problem. *Computers & Operations Research*, 37, pp. 1912-1923, 2010.

- *Université des Andes - Bogota (Colombie)* avec A. L. Medaglia et N. Velasco

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. A matheuristic for the Truck and Trailer Routing Problem. *European Journal of Operational Research*, en révision.

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 38(9), pp. 1319-1334, 2011.

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5), pp 780-794, 2010.

- *Université de Valencia - Valence (Espagne)* avec J.M. Belenguer et E. Benavent

J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. Wolfler-Calvo. A Branch-and-Cut method for the Capacitated Location-Routing Problem. *Computers & Operations Research*, 38, pp. 931-941, 2011.

- *CIRRELT - Montréal (Canada)* avec P. Soriano et A. Ruiz

C. Prins, C. Prodhon, P. Soriano, A. Ruiz, R. Wolfler Calvo, Solving the Capacitated Location-Routing Problem by a Cooperative Lagrangean Relaxation-Granular Tabu Search Heuristic, *Transportation Science*, 41(4), pp. 470-483, 2007.

Séjours de recherche

Juillet 2005 : **Etude de méthodes de coupes pour le problème de localisation-routage**
avec J.M. Belenguer et E. Benavent
Université de Valencia - Valencia (Espagne)

Février - Mars 2005 : **Etude de méthodes exactes et heuristiques pour le problème de localisation-routage**
avec P. Soriano et A. Ruiz
Centre de Recherche sur le Transport - Montréal (Canada)

2.4.3.3 Participation à l'organisation de conférences

Présidente du comité d'organisation d'une conférence de l'EURO working group on metaheuristics dont le thème portait sur l'optimisation logistique et les problèmes liés au domaine du transport : EU/MEeting 2008, Troyes, FRANCE, 23-24 octobre , 2008.

Membre du **comité d'organisation** d'une conférence internationale: IEEE/SSSM'06, Troyes, 2006.

Présidente de sessions pour des conférences internationales : CIE39 (Troyes, 2009), EURO'10 (Lisbonne, 2010), IESM (Metz, 2010).

2.4.3.4 Edition d'un numéro spécial de revue internationale

Suite à l'organisation de la conférence EU/MEeting 2008, nous avons édité un numéro spécial dans la revue Computers & Operations Research :

C. Prins, N. Labadi, C. Prodhon, R. Wolfler Calvo, Metaheuristics for logistics and vehicle routing, *Computers & Operations Research*, Volume 37, Issue 11, 2010.

2.4.3.5 Appartenance à des sociétés et groupes de recherche

EURO : Association of European Operational Research Societies

EU/ME : EURO working group, European chapter on metaheuristics

VeRoLog : EURO Working Group, Vehicle Routing and Logistics Optimization

ROADeF : Société française de recherche opérationnelle et d'aide à la décision

GDR MACS/STP : Groupe de recherche du CNRS en modélisation, analyse et conduite des systèmes, pôle sciences et techniques de la production

GT2L : Groupe de travail du GdR Recherche Opérationnelle du CNRS sur le transport et la logistique

2.4.3.6 Séminaire/Workshop

Journées du GT2L, novembre 2009, Le Havre, France.

Journées STP du GdR MACS, novembre 2006, Valenciennes, France.

CIRRELT, février 2005, Montréal, Canada.

2.5 Thématiques de recherche

Ma recherche se situe dans le domaine de la Recherche Opérationnelle, plus particulièrement en Optimisation Combinatoire avec une prédilection pour les tournées de véhicules. Les méthodes de résolution sont des approches exactes ou heuristiques.

2.5.1 Tournées de véhicules

Ma recherche aborde différents types de problèmes, mais se dirige essentiellement autour des tournées de véhicules ou plus généralement du transport.

Le problème de tournées de véhicules (vehicle routing problem - VRP) consiste à déterminer les itinéraires pour une flotte de véhicules de capacité limitée afin de visiter un ensemble de clients. Le but est de minimiser le coût de transport. Il s'agit d'une extension classique du problème du voyageur de commerce, qui fait partie de la classe des problèmes NP-difficiles. Le VRP a été intensivement étudié durant les cinquante dernières années. En effet, il présente des intérêts aussi bien théoriques que pratiques.

Des extensions de ce problème sont fréquentes afin de s'approcher des réalités du monde industriel. Il est courant par exemple d'ajouter des fenêtres de temps pour la visite des clients, ou de faire partir les tournées d'un ensemble de dépôts. Mon intérêt se porte plus particulièrement sur des cas comprenant un aspect localisation et certaines contraintes de ressources. Bien entendu, la flotte de véhicules à capacité limitée fait déjà apparaître ce type de contraintes, mais l'intérêt ici est d'ajouter d'autres limitations comme sur nombre de véhicules disponibles ou sur les capacités de stockage chez les clients.

Le premier problème étudié est celui concernant la considération simultanée de la localisation et du routage (LRP - Location-Routing Problem). Dans ce problème, il faut déterminer le nombre et la localisation des dépôts qui serviront de point de départ pour les tournées de véhicules qui visiteront les clients. Contrairement au multi-dépôt VRP, il ne s'agit pas seulement de partitionner les clients sur des tournées pouvant débuter de divers endroits. L'utilisation d'un site engendre ici un coût d'ouverture qui contre-balance une potentielle économie dans le cheminement des véhicules. De plus, la variante étudiée comprend des contraintes de capacité limitée à la fois sur les véhicules et les dépôts, ce qui engendre un double problème de Bin-packing lié à l'affectation des clients aux tournées et dépôts. En extension, une partie des travaux porte sur un cas multi-période du problème de localisation-routage.

Ensuite, un autre type de restriction de ressources est initiateur de recherche : la limitation de la flotte de véhicules, en particulier quand celle-ci est hétérogène (Heterogeneous VRP - HVRP). Au problème classique de tournées s'ajoute alors aussi un problème de partitionnement et d'affectation des clients sur des ressources (type de véhicules) présentes en quantité limitée.

L'étude du problème de tournées avec remorques permet de combiner l'idée de flotte hétérogène et de localisation. Il ajoute de plus la notion de tournées multi-niveaux. C'est donc tout naturellement que la poursuite des recherches porte sur les problèmes de localisation-routage à deux niveaux.

Dans la continuité des extensions, après avoir étudié le LRP et sa version périodique, ainsi que le HVRP, des travaux portent sur leurs variantes incorporant une autre gestion de ressources : le stockage, avec la possibilité de servir les clients en avance à condition de respecter les disponibilités d'entreposage.

Enfin, quelques travaux concernent d'autres problématiques comportant des critères antagonistes à optimiser et menant à des études multi-objectifs.

2.5.2 Approches de résolution

La résolution des problèmes de tournées fait appel aux techniques de recherche opérationnelle et d'optimisation combinatoire. Les variantes étudiées peuvent toutes se ramener au VRP classique en cas particulier. Ce sont donc des problèmes NP-difficiles qui ne peuvent être résolus optimalement en des temps de calcul raisonnables, du moins pour les instances de grande taille.

L'optimisation linéaire en nombres entiers permet tout de même de résoudre de façon exacte certains problèmes de tournées de véhicules de taille relativement réduite. De plus, il est intéressant d'utiliser des approches exactes pour développer des bornes et ainsi mieux évaluer la qualité des solutions obtenues heuristiquement. Ainsi, des approches essentiellement de branchement et coupes (Branch and Cut) sont développées.

Cependant, mes recherches sont plutôt basées sur des méthodes approchées. En premier lieu, il s'agit d'heuristiques constructives spécifiques aux problématiques traitées, et permettant d'élaborer des solutions réalisables. En particulier, une version généralisée de l'algorithme de Clarke & Wright est proposée afin de résoudre le cas multi-dépôts. Des adaptations de l'utilisation de l'algorithme Split sont également présentées. Les solutions obtenues sont améliorées à l'aide de recherches locales dédiées aux problèmes étudiés.

Pour échapper aux optima locaux obtenus par les heuristiques suivies de recherches locales, de nombreuses métaheuristiques sont appliquées, comme des algorithmes génétiques, recherches à voisinages variables, etc. Cependant, une autre option concerne des versions plus pertinentes hybridant plusieurs métaheuristiques ou combinant une résolution exacte et heuristique à travers des matheuristiques.

Enfin, il est intéressant de tirer parti de l'architecture des ordinateurs récents, équipés de processeurs multi-cœurs. Ainsi, une implémentation de métaheuristique est développée de façon à exploiter de manière coopérative des explorations parallèles de l'espace des solutions.

Le fil conducteur des approches de résolution proposées est une volonté de garder une vision globale et de proposer une exploration judicieuse de l'espace de recherche, en passant par exemple par des problèmes simplifiés ou relaxés, mais tout en garantissant une certaine coopération pour préserver une efficacité de résolution.

2.6 Liste des publications scientifiques

Je suis l'auteur principal (corresponding author) des références marquées *.

2.6.1 Chapitres de livre

*C. Duhamel, C. Gouinaud, P. Lacomme, C. Prodhon. A multi-thread GRASPxELS for the heterogeneous capacitated vehicle routing problem, dans *Hybrid Metaheuristics*, Springer, en révision.

A. Dolgui, M.-A. Louly, C. Prodhon, A survey on supply planning under uncertainties in MRP environments, dans *Selected Plenaries, Milestones and Surveys* (16th IFAC World Congress, 3-8/07/2005, Prague, République Tchèque), P. Horacek, M. Simandl, and P. Zitech (éd.), pp. 228-239.

2.6.2 Revues internationales

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. A matheuristic for the Truck and Trailer Routing Problem. *European Journal of Operational Research*, en révision.

C. Duhamel, P. Lacomme, C. Prodhon. A hybrid Evolutionary Local Search with depth first search split procedures for the heterogeneous vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 25(2), pp. 345-358, 2012.

V.-P. Nguyen, C. Prins, C. Prodhon. A multi-start Iterative Local Search with tabu list and Path Relinking for the Two-Echelon Location Routing Problem. *Engineering Applications of Artificial Intelligence*, 25(1), pp. 56-71, 2012.

V.-P. Nguyen, C. Prins, C. Prodhon. Solving the Two-Echelon Location Routing Problem by a hybrid GRASPxPath Relinking complemented by a learning process. *European Journal of Operational Research*, 216, pp. 113-126, 2012.

J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. Wolfler-Calvo. A Branch-and-Cut method for the Capacitated Location-Routing Problem. *Computers & Operations Research*, 38, pp. 931-941, 2011.

*C. Duhamel, P. Lacomme, C. Prodhon. Efficient frameworks for greedy split and new depth first search split procedures for routing problems. *Computers & Operations Research*, 38, pp. 723-739, 2011.

*C. Prodhon. A hybrid evolutionary algorithm for the Periodic Location-Routing Problem. *European Journal of Operational Research*, 210(2), pp. 204-212, 2011.

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 38(9), pp. 1319-1334, 2011.

*C. Duhamel, P. Lacomme, C. Prins, C. Prodhon. A GRASPxELS Approach for the capacitated Location-Routing Problem. *Computers & Operations Research*, 37, pp. 1912-1923, 2010

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5), pp 780-794, 2010.

*C. Prodhon. Solving the capacitated location-routing problem. *4OR - A Quarterly Journal of Operations Research*, 5, pp. 339-342, 2007.

*C. Prins, C. Prodhon, P. Soriano, A. Ruiz, R. Wolfler Calvo. Solving the Capacitated Location-Routing Problem by a Cooperative Lagrangean Relaxation-Granular Tabu Search Heuristic, *Transportation Science*, 41(4), pp. 470-483, 2007.

*A. Dolgui, C. Prodhon, Supply planning under uncertainties in MRP environments: a state of the art, *Annual Review on Control*, 31(2), pp. 269-279, 2007 (IF: 0.822).

*C. Prins, C. Prodhon, R. Wolfler Calvo. Solving the Capacitated Location-Routing Problem by a GRASP complemented by a Learning Process and a Path Relinking, *4OR - A Quarterly Journal of Operations Research*, 4(3), pp. 221-238, 2006.

*C. Prins, C. Prodhon, R. Wolfler Calvo. Two-phase Method and Lagrangian Relaxation to Solve the Bi-Objective Set Covering Problem, *Annals of Operations Research*, 147, pp. 23-41, 2006.

2.6.3 Conférences internationales avec comité de lecture et actes publiés

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya. Hybrid Heuristic for the Inventory Location-Routing Problem, *Odysseus 2012* (5th *International Workshop on Freight Transportation and Logistics*), Mykonos, Grèce, 05/2012 (taux de sélection de 40%).

V.-P. Nguyen, C. Prins, C. Prodhon, J.G. Villegas. A branch-and-cut algorithm for the two-echelon location-routing problem, *Odysseus 2012* (5^{th} *International Workshop on Freight Transportation and Logistics*), Mykonos, Grèce, 05/2012 (taux de sélection de 40%).

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. Heuristic column generation for the truck and trailer routing problem. *IESM 2011*, (International Conference on Industrial Engineering and Systems Management, Metz, 25-27/05/2011).

V.-P. Nguyen, C. Prins, C. Prodhon. A hybrid metaheuristic for the two-echelon location routing problem. *IESM 2011*, (International Conference on Industrial Engineering and Systems Management, Metz, 25-27/05/2011).

V.-P. Nguyen, C. Prins, C. Prodhon. A Multi-Start Evolutionary Local Search for the Two-Echelon Location Routing Problem. dans: *Hybrid Metaheuristics* (actes de Hybrid Metaheuristics 2010, Octobre 2010, Vienne, Autriche), M.J. Blesa et al. (éd.), Lecture Notes in Computer Science, Springer, 2010.

*C. Prodhon. An ELSxPath Relinking Hybrid for the Periodic Location-Routing Problem, dans *Hybrid Metaheuristics* (actes de Hybrid Metaheuristics 2009, Octobre 2009, Udine, Italie), M.J. Blesa *et al.* (éd.), Lecture Notes in Computer Science 5818, pp. 15-29, Springer, 2009. ISBN 978-3-540-88438-5 (**taux de sélection de 40%**).

C. Prodhon, D. Macias, F. Yalaoui, A. Vial, L. Amodeo. Bi-Objective Optimization of the Plasmon-Assisted lithography - Design of plasmonic nanostructures. *International joint conference on computational intelligence*, Funchal, Portugal, 05-07 October 2009. 4 p.

*C. Prodhon. An evolutionary algorithm for the periodic location-routing problem, *Odysseus 2009* (4^{th} *International Workshop on Freight Transportation and Logistics*), ζesme, Turquie, 23-26/05/2009, Actes sur CD.

C. Prodhon, D. Macias, F. Yalaoui, A. Vial, L. Amodeo. Evolutionary optimization for Plasmon-assisted lithography. dans: *EvoWorkshops 2009*: Applications of Evolutionary Computing (actes d'EvoIASP 2009, Tübingen, Germany, April 15-17, 2009), M. Giacobini et al. (éd.), Lecture Notes in Computer Science 5484, pp. 420-425, Springer Verlag, 2009. ISBN 978-3-642-01128-3.

*C. Prodhon, C.Prins.A Memetic Algorithm with Population Management(MA|PM) for the Periodic Location-Routing Problem, dans *Hybrid Metaheuristics* (actes de Hybrid Metaheuristics 2008, Septembre 2008, Málaga, Espagne), M.J. Blesa *et al.* (éd.), Lecture Notes in Computer Science 5296, pp. 43-57, Springer, 2008. ISBN 978-3-540-88438-5 (**taux de sélection de 40%**).

*C. Prodhon. A Metaheuristic for the Periodic Location-Routing Problem, *GOR'07* (*Operations Research 2007*), Saarbrücken, Allemagne, 5-7/09/2007. Springer, 2008, 6 p.

*J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. Wolfler-Calvo. A branch-and-cut method for the Capacitated Location-Routing Problem, *IEEE/ICSSSM'06* (*IEEE International Conference on Services Systems and Services Management*, Troyes, France, 25-27/10/2006). Actes sur CD édités par IEEE, 2006. ISBN 1-4244-0451-7.

*J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. Wolfler-Calvo. A Cutting Plane method for the Capacitated Location-Routing Problem, *Odysseus 2006* (*Third International Workshop on Freight Transportation and Logistics*, Altea, Espagne, 23-26/05/2006), E. Benavent et al. (éd.), pp. 50-52, Université de Valencia, Valencia, 2006. ISBN 84-96221-33-4.

*C. Prins, C. Prodhon, R. Wolfler Calvo. A Memetic Algorithm with Population Management (MA|PM) for the Capacitated Location-Routing Problem, dans *Evolutionary computation in combinatorial optimization* (actes d'EvoCOP 2006, avril 2006, Budapest, Hongrie), J. Gottlieb et G.R. Raidl (éd.), Lecture Notes in Computer Science 3906, pp. 183-194, Springer, 2006. ISBN 3-540-33178-6 (**taux de sélection de 30%**).

*C. Prins, C. Prodhon, R. Wolfler Calvo. A reactive GRASP and Path Relinking algorithm for the Capacitated Location Routing Problem, *IESM'05*, (*International Conference on Industrial Engineering and Systems Management*, Marrakech, 16-19/05/2005), 10 pages. Actes sur CD édités par I4E2, Mons, Belgique, 2005. ISBN 2-9600532-0-6.

2.6.4 Conférences internationales avec comité de lecture sans actes publiés

V.-P. Nguyen, C. Prins, C. Prodhon. A cutting plane method for the Two-Echelon Location Routing Problem. *Optimization 2011*, Lisbon, Portugal, 2011.

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. Heuristic column generation for the truck and trailer routing problem. *ROUTE 2011*, Sitges, Spain, 01-03/06/2011.

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya. A Matheuristic for the Inventory Location Routing Problem with Deterministic Demand. *MIC 2011* (*Metaheuristics International Conference*), Udine, Italie, 2011.

W.J. Guerrero, N. Velasco, C.A. Amaya , C. Prodhon. The Inventory Location Routing Problem with Deterministic Demand. *POM 2011* (*22nd Annual Conference of the Production and Operations Management Society*), 2011.

C. Duhamel, P. Lacomme, C. Prodhon, C. Prins. Parallel Cooperative GRASP for the Heterogeneous Vehicle Routing Problem. *EURO XXIV* (*European Conference On Operational Research*), Lisbon, Portugal, July 11-14, 2010.

V.-P. Nguyen, C. Prins, C. Prodhon. A hybrid GRASP x Path Relinking for the Two-Echelon Location Routing Problem. *EURO XXIV* (*European Conference On Operational Research*), Lisbon, Portugal, July 11-14, 2010.

*C. Prodhon. Solution methods for the periodic location-routing problem. *EURO XXIV* (*European Conference On Operational Research*), Lisbon, Portugal, July 11-14, 2010.

J.G. Villegas, J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon. A cutting plane approach for the single truck and trailer routing problem with satellite depots. *EURO XXIV* (*European Conference On Operational Research*), Lisbon, Portugal, July 11-14, 2010

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. GRASP/VND with path relinking for the truck and trailer routing problem. *TRISTAN VII* (7^{th} *triennial symposium on transportation analysis*), Tromsø, Norway, June 20-25, 2010.

P. Lacomme, C. Prodhon. Parallel Cooperative ELS for the Heterogeneous Vehicle Routing Problem. *EU/MEeting 2010* (10^{th} *Anniversary of the Metaheuristics Community*), Lorient, France, 03-04/06/2010.

*C. Duhamel, P. Lacomme, C. Prins, C. Prodhon. A GRASP×ELS approach for real-life Location Routing Problems. *CIE39 (*International Conference on Computers & Industrial Engineering*)*, Troyes, France, 06-08/07/2009.

J.-G. Villegas, A. Medaglia, C. Prins, C. Prodhon, N. Velasco. Metaheuristics for a truck and trailer routing problem. *CORAL 2009* (5^{th} *Conference on Routing and Location*), Elche, Espagne, 10-12/06/2009.

J.-G. Villegas, A. Medaglia, C. Prins, C. Prodhon, N. Velasco. GRASP/Evolutionary Local Search Hybrids for a Truck and Trailer Routing Problem, *MIC 2009* (*8th Metaheuristics International Conference*), Humbourg, Allemagne, 13-16/07/2009.

*C. Duhamel, P. Lacomme, C. Prins, C. Prodhon. A Memetic Approach for the Capacitated Location Routing Problem. *EU/MEeting 2008* (*International Workshop on Metaheuristics for Logistics and Vehicle Routing*), Troyes, France, 23-24/10/2008.

J.-G. Villegas, A. Medaglia, J. Mendoza, C. Prins, C. Prodhon, N. Velasco. A split-based framework for the vehicle routing problem. *CLAIO 2008* (*XIV Congreso Latino Ibero Americano de Investigacion de Operaciones*), Catagena, Colombie, 9-12/09/2008.

*J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. Wolfler Calvo. Some computational results with integer formulations for the Location-Routing Problem. *CORAL 2006* (*4th Conference on Location and Routing*), Puerto de la Cruz (Tenerife), Espagne, 14-16/10/2006.

*J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. Wolfler-Calvo. New results with integer formulations for the Capacitated Location-Routing Problem, *EURO XXI* (*21st European Conference on Operational Research*), Reykjavik, Islande, 02-

05/07/2006.

*C. Prins, C. Prodhon, P. Soriano, A. Ruiz, R. Wolfler Calvo. A cooperative Lagrangean Relaxation-Granular Tabu Search heuristic for the Capacitated Location Routing Problem, *MIC 2005* (*6th Metaheuristics International Conference*), Vienna, Autriche, 22-26/08/2005. Actes sur CD, Université de Vienne, pp. 769-774.

2.6.5 Conférences nationales avec comité de lecture et actes publiés

V.-P. Nguyen, C. Prins, C. Prodhon. GRASP with learning process for a Two-Echelon Location Routing Problem. *MOSIM'10* (*8ème Conférence Francophone de Modélisation et Simulation*, Hammamet, Tunisie, 10-13/05/2010), 10p.

*C. Prins, C. Prodhon, R. Wolfler Calvo. Nouveaux Algorithmes pour le Problème de Localisation et Routage sous Contraintes de Capacité, *MOSIM'04* (*4ème Conférence Francophone de Modélisation et Simulation*, Nantes, France, 01-03/09/2004), pp. 1115-1122, Editions Tec-Doc Lavoisier, 2004. ISBN 2-7430-0731-1.

2.6.6 Conférences nationales avec comité de lecture sans actes publiés

C. Duhamel, P. Lacomme, C. Prodhon. Problème de tournées de véhicules à flotte hétérogène limitée et gestion des stocks intégrée. *Congrès ROADEF 2012*, Angers, France, 04/2012.

W. Guerrero, C. Prodhon, N. Velasco, C.-A. Amaya. Integration d'une gestion de stocks lors de la résolution coopérative du problème de Localisation-Routage. *Congrès ROADEF 2012*, Angers, France, 04/2012.

V.F. Nguyen, C. Prins, C. Prodhon. Approches heuristiques pour le Problème de Localisation - Routage à deux niveaux. *Congrès ROADEF 2010*, Toulouse, France, 03/2010.

C. Duhamel, P. Lacomme, C. Prodhon. Définition d'un Split en Profondeur pour les problèmes de tournées. *Congrès ROADEF 2010*, Toulouse, France, 03/2010.

*C. Prodhon. An ELSxPath Relinking Hybrid for the Periodic Location-Routing Problem, *Journée du GT Transport et Logistique*, 6/11/2009, Le Havre, France.

*C. Prodhon. Etude du problème de localisation-routage périodique. *Congrès ROADEF 2008*, Clermont-Ferrand, France, 02/2008.

*C. Prodhon. Approche Coopérative pour le Problème de Localisation-Routage, *Journées STP du GdR MACS*, 16/11/2006, Valenciennes, France.

*C. Prins, C. Prodhon, R. Wolfler Calvo. Comparaisons de Méthodes de Résolution pour le Problème de Localisation-Routage sous Contraintes de Capacité, *Congrès ROADEF 2006*, Lille, France, 06-08/02/2006, résumé étendu de 2 pages sur CD.

*C. Prins, C. Prodhon, P. Soriano, A. Ruiz, R. Wolfler Calvo. A metaheuristic for the Capacitated Location Routing Problem Combining Lagrangean Relaxation with Granular Tabu Search heuristic, *AIRO2005* (*XXXVI° Conferenza annuale Associazione Italiana Ricerca Operativa*), Camerino, Italie, 6-9/09/2005.

2.6.7 Autres

Mémoire de thèse

*C. Prodhon. Le Problème de Localisation-Routage, Université de Technologie de Troyes, octobre 2006, Troyes, France.

Séminaire

*C. Prodhon. A reactive GRASP and Path Relinking algorithm for the Capacitated Location Routing Problem, Centre de Recherche sur les Transports, février 2005, Montréal, Canada.

Partie 3

Synthèse des Activités de Recherche

.

3.1 Introduction

Mes activités de recherche à l'institut Charles Delaunay de l'université de technologie de Troyes, au sein du laboratoire d'optimisation des systèmes industriels (LOSI), s'inscrivent dans le domaine de la logistique du transport, et plus précisément de l'optimisation de problèmes de tournées de véhicules.

La logistique est l'activité relative aux méthodes et aux moyens d'organisation d'une opération, d'un processus. A l'origine dédiée aux opérations ayant pour but de combiner le transport et le ravitaillement des armées, elle a aujourd'hui plus généralement pour objet de gérer les flux physiques d'une organisation, dans le but de mettre à disposition les ressources correspondant aux besoins. Quant au transport, il consiste à déplacer quelque chose ou quelqu'un d'une origine vers une destination. Pour cela, un mode de transport est utilisé (souvent un véhicule) sur un réseau (routier, fluvial, ferroviaire, électrique ...).

L'activité de transport s'est fortement développée depuis la révolution industrielle, avec la mondialisation du marché, les délocalisations et le développement du tourisme. D'après une évaluation de l'Organisation Mondiale du Commerce, la part des échanges internationaux dans le PIB mondial est passée de 5,5% en 1950 à 20,5% en 2006. Ainsi les transports affectent de plus en plus l'environnement global et local avec la pollution lumineuse, sonore et atmosphérique entre autres. L'agence internationale de l'énergie (AIE) a estimé qu'en 2004, le transport était responsable de 23% des émissions de gaz à effet de serre mondiales en rapport avec l'énergie, et 74% des émissions de CO2 en rapport avec l'énergie dans le secteur des transports viennent des transports routiers (OMC, 2012).

Ainsi, les coûts reliés au transport occupent aujourd'hui une place non-négligeable, en particulier dans l'industrie, et la logistique qui y est reliée devient quant à elle un des facteurs majeurs de la compétitivité des entreprises, tant pour la maîtrise des coûts que pour celle des niveaux de service. C'est pourquoi les entreprises ont tout intérêt à s'intégrer dans des réseaux d'approvisionnement et de distribution pour tenter de réduire au plus leurs coûts. Il faut savoir par exemple que la distribution/collecte pour le dernier kilomètre est la plus coûteuse, environ 20% du coût total de la chaîne (PIPAME, 2009).

La logistique du transport soulève un grand nombre de problèmes, très souvent difficiles à résoudre de manière optimale. En particulier, l'optimisation des tournées de véhicules suscitent un grand intérêt, que cela soit dans le monde académique ou industriel, puisqu'elle peut s'appliquer à de nombreux domaines, aussi bien le transfert de produits physiques avec la distribution du courrier, la livraison de colis, la collecte de lait, que l'agencement de lignes de transports en commun, le salage du réseau routier, le ramassage des déchets ménagers, etc.

Cependant, des efforts sont encore nécessaires afin de développer des systèmes de transport flexibles et efficaces répondant aux préoccupations actuelles relatives à l'économie et à notre qualité de vie. La Recherche Opérationnelle, plus spécialement l'Optimisation Combinatoire, sur ce type de problèmes s'avère donc plus que jamais essentielle.

Dans la suite, tout d'abord les différentes problématiques étudiées seront introduites, généralement relatives aux tournées de véhicules agrémentées de contraintes supplémentaires. Ensuite, les méthodologies développées, basées sur des approches heuristiques ou exactes de résolution, seront exposées.

3.2 Problématiques étudiées

Comme cela a été présenté en introduction, mes travaux de recherches s'inscrivent principalement dans l'optimisation de la logistique du transport et plus précisément des problèmes de tournées de véhicules (Toth et Vigo, 2001). Ces derniers consistent à déterminer les tournées d'une flotte de véhicules, disposant d'une capacité limitée, afin de visiter un ensemble de clients. Une hypothèse courante est de supposer qu'un client ne doit être visité que par un seul véhicule et une seule fois. Le but est de minimiser le coût de transport. Une manière de les modéliser repose sur l'utilisation d'un graphe. Cette représentation est en effet parfaitement adaptée pour définir un réseau quelconque, comme un réseau routier ou électrique.

Pour cela, il est nécessaire de définir quelques éléments. Soit :

- V un ensemble de sommets ou *nœuds*.
- A une famille de paires ordonnées de sommets appelées *arcs*.

En pratique un nœud i peut correspondre à un emplacement de site (usine, dépôt, plate-forme, ...), mais aussi à un carrefour, et un arc (i, j) symbolise un tronçon de communication comme une route reliant deux carrefours, avec un sens de circulation.

Une distinction possible est sur la notion de *client* à visiter. Il peut s'agir soit d'un arc soit d'un nœud. Les travaux réalisés s'intéressent plus spécifiquement à un ensemble de clients représentés par des nœuds.

Si toutes les rues sont à double sens, l'existence d'un arc (i, j) de i vers j implique celle de l'arc opposé (j, i), et le graphe est dit symétrique. Dans ce cas, on peut utiliser un modèle de graphe non-orienté $G = (V, E)$, dans lequel l'ensemble A est remplacé par un ensemble E d'arêtes (liaisons non-orientées).

Les tronçons peuvent être pondérés, ou valués, c'est-à-dire que chaque arc du graphe est muni d'un poids, ou *coût*, défini par une application C de A dans $\mathbb{R}$. Le graphe se note alors $G = (V, A, C)$. Concrètement, dans un réseau routier par exemple, les valuations peuvent correspondre à des temps de parcours, des distances, des capacités en nombre de véhicules/heure, une limitation de vitesse, etc.

Les deux principaux problèmes académiques de tournées sont alors :

- *TSP ou Traveling Salesman Problem.* C'est le problème du voyageur de commerce, où tous les clients doivent être visités en une seule tournée.
- *VRP ou Vehicle Routing Problem.* C'est le problème classique de tournées de véhicules. Chaque client j a une demande connue d_j. Les véhicules ont une capacité limitée Q qui oblige à faire plusieurs tournées pour satisfaire les demandes.

Bien que le TSP soit le problème de tournées sur nœuds le plus simple, il est déjà *NP*-difficile (Johnson et Papadimitriou, 1985). C'est également le cas de toutes les variantes des problématiques de tournées exposées dans ce rapport.

Par cette représentation sous forme de graphe, diverses extensions du problème de tournées de véhicules sont abordées ici. En particulier, le premier problème décrit dans la suite intègre un cas ajoutant un aspect de localisation de dépôts. Outre le choix de localisation, une limitation de l'espace de stockage des dépôts est considérée afin d'être plus réalistes pour les entreprises.

3.2.1 Le problème de localisation-routage

Quand la capacité d'un unique dépôt n'est pas suffisante pour satisfaire l'ensemble des clients, il est alors nécessaire d'en ouvrir plusieurs. Il faut donc définir à partir de quel dépôt chacun des clients sera servi (problème d'affectation) en plus du problème d'élaboration des tournées. Le problème devient alors le problème de tournées multi-dépôts. Cependant, si l'on ajoute le fait que ces dépôts ne sont pas encore ouverts, et qu'il va falloir décider de leur nombre et de leur localisation (chacun engendrant potentiellement un coût d'ouverture), le problème devient le problème de localisation-routage (*LRP - Location-Routing Problem*). Cette problématique a fait l'objet de mes travaux de thèse soutenue en 2006, poursuivis en collaboration avec une équipe du LIMOS pour aboutir à un chapitre de livre (Duhamel *et al.*, à paraître), cinq articles de revues internationales (Prins *et al.*, 2006c, 2007; Prodhon, 2007; Duhamel *et al.*, 2010a; Belenguer *et al.*, 2011) et treize présentations en conférences (Prins *et al.*, 2004, 2005a,b,c, 2006a,b; Prodhon, 2006; Belenguer *et al.*, 2006a,b,c,d; Duhamel *et al.*, 2008, 2009).

A première vue, résoudre un *LRP* peut sembler peu réaliste puisqu'il combine des décisions à long terme (décisions stratégiques concernant les dépôts) et à plus court terme (décisions tactiques et/ou opérationnelles concernant les tournées). Souvent les futurs clients et leur demande sont mal connus au moment du choix de l'emplacement des dépôts. Dans de telles conditions, le *LRP* peut être peu adapté. Par contre, dans les configurations où les tournées sont stables, comme en maintenance (relevé de compteurs) ou en collecte de déchets ménagers, la prise en compte des futures tournées au moment de la localisation des dépôts devient plus justifiée. De même, les coûts fixes d'ouverture d'un site peuvent être en fait des coûts d'exploitation pour une période donnée (location d'entrepôts existants par exemple). Ainsi, ces derniers deviennent d'un ordre de grandeur comparable aux coûts des tournées de la période considérée. Enfin, on peut citer le cas des secours humanitaires, où il faut situer temporairement des bases de ravitaillement et élaborer des tournées d'approvisionnement pour venir en aide aux populations dans le besoin. Dans tous ces exemples, le fait de prendre en compte le coût des futures tournées rend le LRP plus précis que les problèmes de localisation qui approximent les coûts de transport par des livraisons directes.

La résolution du *LRP* représente donc un intérêt non négligeable dans divers cas de figure. Min *et al.* (1998) proposent une classification du LRP en tenant compte des contraintes de capacités sur les dépôts, des caractéristiques de la flotte de véhicules (homogène ou hétérogène) et des coûts d'utilisation des véhicules. Cependant, le nombre de travaux sur le sujet est relativement peu élevé, en particulier pour des cas intégrant des capacités limitées à la fois sur les véhicules et les dépôts (voir l'état de l'art de Nagy et Salhi (2007)). Pourtant, cette version, appelée *LRP* généralisé ou Capacitated LRP (CLRP), est plus réaliste en logistique du transport et c'est donc celle qui est abordée ici. Jusqu'en 2004, il n'y avait guère que Wu *et al.* (2002) qui avait traité le CLRP mais avec une flotte homogène ou hétérogène limitée.

Plus formellement, le CLRP consiste d'une part à déterminer l'ouverture d'un certain nombre de dépôts i, de capacité limitée W_i, parmi un sous-ensemble I possible, et d'autre part à élaborer des tournées de véhicules de manière à visiter un ensemble J de n clients. Chaque client j a une demande d_j spécifique, et chaque tournée est effectuée par un véhicule, de capacité limitée Q. Le but est de minimiser le coût total, comprenant le coût d'ouverture des dépôts O_i, le coût fixe d'utilisation d'un véhicule F et la somme des coûts c_{ij} des arêtes traversées par les véhicules.

Dans le passé, lors de la résolution de ce type de problème, la démarche majoritairement employée était de traiter chaque niveau de décision séparément, en commençant par le niveau stratégique. Cependant, les approximations alors nécessaires engendrent souvent des solutions moins avantageuses en terme de coût. C'est d'ailleurs ce qui est constaté lorsque l'on sépare les décisions de localisation de dépôts et celles concernant l'élaboration des tournées de véhicules (Salhi et Rand, 1989). Pour cette raison, une étude du problème dans sa globalité est privilégiée.

L'objectif étant essentiellement de résoudre des instances de taille réaliste, les heuristiques développées pour ce problème, et qui seront exposées dans la partie dédiées aux méthodologies 3.3.2, sont relativement rapides par rapport aux métaheuristiques. Il s'agit de méthodes constructives et de recherches locales.

Basée sur ces heuristiques, une série de métaheuristiques est proposée afin d'obtenir des résultats plus pertinents :

- une procédure de recherche adaptative gloutonne et randomisée (appelée *GRASP*). Afin de la rendre plus pertinente, la méthode de base est ici renforcée par une technique d'apprentissage et par une post-optimisation par Path Relinking (*PR*) (Prins *et al.*, 2006c).

- un algorithme mémétique avec gestion de la population (Prins *et al.*, 2006b).
- une approche dite *coopérative*, plus précisément une matheuristique. Cette méthode est basée sur un rapport non hiérarchique entre des phases de localisation et de constitution de tournées. Ainsi, le choix des dépôts repose sur une formulation simplifiée du problème de départ, permettant d'obtenir un problème de localisation résolu à l'aide d'une relaxation lagrangienne avec une optimisation par sous-gradients et une heuristique lagrangienne. La partie sur les tournées est améliorée par une métaheuristique de type tabou granulaire (Prins *et al.*, 2007).
- une méthode hybride combinant un GRASP avec une recherche locale évolutionnaire (*ELS - Evolutionary Local Search*) (Duhamel *et al.*, 2010a)

Plus de détails sur ces méthodes sont donnés dans les parties 3.3.3, en particulier l'approche coopérative très performante rappelée au 3.3.3.3.

Enfin, un modèle détaillé dans la partie 3.3.1, une méthode exacte et des bornes inférieures sont développés. La résolution se base alors sur un algorithme de coupes (Belenguer *et al.*, 2011).

3.2.2 Le problème de tournées de véhicules à flotte hétérogène

Un autre type de problème de tournées avec contraintes additionnelles sur les ressources disponibles peut concerner la limitation de la flotte de véhicules, en particulier quand celle-ci est hétérogène. Au problème classique de tournées s'ajoute alors, comme dans le LRP, un problème de partitionnement et d'affectation des clients sur des ressources (type de véhicules) présentes en quantité limitée. Ce second problème d'étude est classiquement appelé le problème de tournées de véhicules à flotte hétérogène (HVRP). Les travaux sur ce sujet sont réalisés en collaboration avec une équipe du LIMOS et publiés sous forme d'un chapitre de livre (Duhamel *et al.*, à paraître), deux articles de revues internationales (Duhamel *et al.*, 2011, 2012a) et deux présentations en conférences (Duhamel *et al.*, 2010c; Lacomme et Prodhon, 2010).

Le problème à flotte hétérogène est relativement similaire à la version classique du VRP. En fait, il le généralise au cas où différents types de véhicules sont disponibles. Plusieurs variantes sont alors envisageables.

Tout d'abord celle pour laquelle chaque type de véhicule est défini par une capacité, un coût fixe et un coût variable de transport (proportionnel à la distance parcourue). Comme pour le VRP, la flotte n'a pas de taille limitée (le nombre de véhicules n'est pas borné). Cette variante est appelée Vehicle Fleet Mix Problem (VFMP), introduite par Golden *et al.* (1984). La première version ne considère que des coûts fixes (VFMP-F) et une première métaheuristique lui est dédiée par Osman et Salhi (1996). Taillard (1999) propose une version avec un coût variable par unité de distance (VFMP-V) et Choi et Tcha (2007) combine les deux coûts (VFMP-FV).

La seconde variante considère un nombre limité de véhicules par type. Ce problème devient ainsi beaucoup plus contraint que le VRP ou les VFMP, puisqu'on y ajoute une limitation de ressources et ceci ne favorise pas leur résolution. On obtient l'Heterogeneous Fleet VRP (HVRP). Dans la plupart des publications, les algorithmes de résolution ne sont dédiés qu'à un seul type de problème de tournées avec flotte non-homogène (VFMP). Récemment néanmoins, quelques articles s'intéressent au HVRP (Li *et al.*, 2010; Brandão, 2011) et Prins (2009b) est le premier à traiter l'ensemble des instances VFMP et HVRP avec le même algorithme.

L'objectif des approches de résolution heuristique développées ici est de traiter toutes les variantes à flotte hétérogène, en particulier celle avec limitation de ressources (HVRP). Pour cela, une procédure Split détaillée en 3.3.2.2 et une Evolutionary Local Search (ELS) sont proposées dans les articles Duhamel *et al.* (2011, 2012a).

3.2.3 Le problème de tournées avec remorques

Concernant les types de véhicules, il est possible d'envisager le cas d'un camion-tracteur et de sa remorque. Le camion-tracteur dispose une capacité minimale de stockage, mais en contre-partie, il est plus petit et peut ainsi servir tous les clients, même les moins accessibles. Quand il accroche sa remorque, il augmente sa capacité de stockage (il ajoute de la ressource), mais l'accès devient alors interdit chez certains clients.

Considérer que les tournées partent et reviennent au dépôt avec une seule configuration de véhicule revient à un problème à flotte hétérogène : une tournée commençant avec la remorque s'effectue entièrement avec cette dernière (premier type de véhicule à grande capacité), les clients n'étant pas accessibles avec la remorque sont alors visités par des tournées à part effectuées par le camion-tracteur seul (second type de véhicule à faible capacité). Cependant, l'idée ici est tout autre. Il s'agit de supposer que le camion qui part avec sa remorque peut déposer cette dernière (sur un parking par exemple), visiter un sous-ensemble de clients, revenir à l'endroit où se trouve la remorque, transférer des produits du camion à la remorque et potentiellement la rattacher, puis repartir. On obtient ainsi des tournées avec sous-tours, qui a priori minimisent le coût de routage puisqu'il n'est pas nécessaire de retourner au dépôt. L'inconvénient est que ce problème devient encore plus complexe puisque s'ajoutent les décisions de localisation des lieux de dépôt de la remorque (parkings). Ce n'est cependant pas un problème de localisation-routage tel que vu précédemment, en partie du fait que les "lieux de dépôt" ne sont pas contraints en capacité et n'ont pas de coût d'utilisation.

Ce problème trouve son application quand les conditions d'accès aux clients sont restrictives. C'est le cas par exemple pour la collecte de lait (Claassen et Hendriks, 2007), où les fermes sont parfois isolées et accessibles par des chemins difficiles. On peut penser aussi aux problèmes postaux, le postier pouvant être assimilé au camion-tracteur et son véhicule à la remorque (Levy et Bodin, 2000), ou aux problèmes de livraison avec contraintes d'accessibilité dans les centre-villes.

Les travaux sur ce sujet sont réalisés en collaboration avec Christian Prins et une équipe l'université des Andes en Colombie par le biais de la thèse en cotutelle de Juan Guillermo Villegas, soutenue en Décembre 2010. Ils ont débouché sur des publications sous forme d'articles de revues internationales (Villegas *et al.*, 2010b, 2011a, en révision) et de présentations en conférences (Villegas *et al.*, 2009a,b, 2010a,c, 2011b,c). Plus précisément, deux variantes ont été étudiées.

3.2.3.1 Le problème de tournée simple avec remorque et dépôts satellites

La variante à tournée simple et dépôts satellites reprend les caractéristiques citées précédemment. Cependant, on suppose qu'un seul véhicule avec sa remorque est disponible et qu'il est capable de satisfaire toute la demande des clients. Les localisations possibles pour détacher la remorque (parkings ou *trailer points*) constituent un ensembles de positions connues. Les clients ne sont accessibles que par le camion sans la remorque.

Pour résoudre ce problème, appelé le Single Truck and Trailer Routing Problem with Satellite Depots, STTRPSD, il faut donc définir les lieux de parking qui seront utilisés et qui représentent les dépôts-satellites, affecter les clients à ces dépôts-satellites puis résoudre le routage. Ce dernier se décompose en deux parties : le premier niveau entre le dépôt principal et les dépôts-satellites, effectué par le camion avec sa remorque, et le second niveau qui consiste à visiter les clients à partir des satellites, tout en respectant la contrainte de capacité du camion-tracteur. Le coût total à minimiser concerne le routage des deux niveaux.

Le problème de tournée simple avec remorque et dépôts-satellites a fait l'objet de la conception de trois heuristiques, deux métaheuristiques basées sur un GRASP et une ELS, et un algorithme exact de branchement et coupes. En particulier, dans la partie 3.3.2.2 sera rappelée une proposition d'algorithme de type Split particulièrement subtile.

3.2.3.2 Le problème de tournées avec remorques

La seconde variante, le Truck and Trailers Routing Problem (TTRP), concerne le problème de tournées avec remorques qui est une version multi-véhicules avec une flotte hétérogène limitée. Cela signifie que le nombre de camions-tracteurs m_t et le nombre de remorques m_r sont limités et $m_t > m_r$.

Cette fois-ci, tous les clients n'ont pas forcément une contrainte d'accessibilité et certains peuvent alors être servis par le camion avec sa remorque. Dans ce cas, il est possible de déposer la remorque chez ces clients. Les parkings disponibles de la première variante sont donc ici les clients sans restriction d'accessibilité.

Le graphe représentant ce problème peut être partitionné en deux sous-ensembles : celui composé des nœuds N_t accessibles par le camion seulement et celui composé des nœuds N_v sans restriction. Une solution peut contenir trois types de tournées : les tournées d'un camion simple (sans remorque) visitant potentiellement les clients de N_v et N_t, les tournées avec la remorque ne visitant que des clients de N_v et les tournées avec sous-tours (avec remorque déposée chez un client N_v pour visiter des clients de N_t). Le coût total à minimiser concerne le routage des deux niveaux.

Ce problème a de nombreuses applications. Semet et Taillard (1993) étudient une variante pour le ravitaillement d'épiceries en Suisse. Ils proposent un algorithme tabou pour ce problème qui comprend également des fenêtres de temps, des dépendances de sites et une flotte hétérogène.

Gerdessen (1996) décrit deux autres applications possibles. La première concerne la distribution de produits laitiers aux Pays-Bas, où l'utilisation des camions avec remorques est courante. Comme dans le STTRPSD, les clients difficiles d'accès (au centre-ville par exemple) ne peuvent être servis par camion complet et la remorque doit être détachée sur un parking. La seconde est la distribution de nourriture pour animaux dans les régions rurales où les clients ne sont accessibles que par de petites routes et/ou des passages limités, comme de petits ponts.

Une autre application est la collecte de lait. Par exemple, Hoff et Løkketangen (2007) proposent une approche par un algorithme tabou avec une modélisation sous forme d'un TTRP multi-dépôts où les clients sont tous de type N_t et les remorques sont déposées sur des parkings.

Le problème de tournées avec remorques a fait l'objet du développement de deux principales méthodes : une métaheuristique hybridant un GRASP, une recherche à voisinage variable et un path relinking, ainsi qu'une matheuristique qui utilise les routes élaborées par le GRASP/VNS pour résoudre le TTRP formulé sous forme d'un problème de partitionnement (voir aussi la partie 3.3.3.3).

3.2.4 Le problème de localisation-routage à deux niveaux

Le problème de tournées avec remorques peut être vu en quelque sorte comme une combinaison des problèmes de localisation-routage et de tournées de véhicules avec flotte hétérogène. Une autre variante de cette combinaison s'obtient en relaxant la contrainte de flotte limitée et en ajoutant d'une contrainte de capacité limitée et un coût fixe associé aux dépôts satellites. Le problème résultant est de type localisation-routage à deux niveaux - LRP-2E (2-Echelon LRP). Ce problème est étudié dans le cadre de la thèse de Viet Phuong Nguyen soutenue en Décembre 2011 et co-encadrée avec Christian Prins. Il en a découlé diverses publications telles que deux articles de revues internationales (Nguyen *et al.*, 2012a,b) et sept présentations en conférences (Nguyen *et al.*, 2010a,b,c,d, 2011a,b, 2012c).

Le LRP-2E est un problème d'optimisation des systèmes de distribution / collecte à deux niveaux dans lesquels des marchandises sont délivrées / collectées entre un dépôt principal et des clients en traversant des dépôts intermédiaires (appelés satellites). Ce problème consiste donc à choisir les satellites à ouvrir et à construire les tournées de véhicules sur les deux niveaux.

Cette fois-ci, l'ensemble des nœuds est composé de trois sous-ensembles : $V_o = \{0\}$ le dépôt principal, $V_s = \{1, 2, \ldots, m\}$ les m satellites possibles et $V_c = \{m+1, m+2, \ldots, m+n\}$ les n clients. Un ensemble K_1 de véhicules de 1^e niveau ayant la même capacité k_1 et un coût fixe F_1 est disponible au dépôt. Un ensemble K_2 de véhicules de 2^e niveau ayant la même capacité k_2 est partagé par les satellites avec un coût fixe F_2. Le nombre de tournées réalisées n'est pas limité. Chaque tournée de 2^e niveau doit partir et revenir au même satellite.

L'objectif est de déterminer l'ensemble des satellites à ouvrir et les tournées des deux niveaux permettant de minimiser le coût total du système.

Les applications concernent notamment la logistique urbaine. En effet, la distribution/collecte pour le dernier kilomètre est la plus coûteuse (environ 20% du coût total de la chaîne) sans compter que certaines municipalités tentent de réduire le traffic dans leurs villes par le biais de restriction d'accès ou de taxation. De plus, le fonctionnement d'un système de transport requiert bien souvent des plates-formes logistiques. Ainsi, on observe la création de ces plates-formes en périphérie des agglomérations, à partir desquelles des véhicules plus petits ont accès au centre-ville. Ceci permet des économies de transport considérables par rapport aux livraisons directes.

La première étude publiée sur le LRP-2E semble être celle de Jacobsen et Madsen (1980) qui porte sur un cas réel de distribution de journaux au Danemark. Ils proposent des heuristiques sans recherche locale, testées sur une seule instance.

Ensuite, les seules méthodes développées sont celles de Boccia *et al.* (2010) qui ont conçu une méthode taboue pour une version avec plusieurs dépôts principaux.

Jusqu'à maintenant, les travaux sur le LRP-2E sont donc rares, mais on peut lister des sous-problèmes s'y apparentant. Par exemple, le problème de tournées multi-niveaux (N-Echelon Vehicle Routing Problem - VRP-NE) décrit une classe de problèmes dans laquelle les tournées de véhicules partent d'un ou plusieurs dépôts pour servir des clients en passant par des dépôts intermédiaires (dit encore satellites ou points de transfert). Dans sa thèse soutenue en 2008, Gonzalez-Feliu (2008) propose un modèle mathématique de type flot de véhicules pour le VRP-2E ainsi qu'une approche exacte par génération de colonnes et branch-and-price. Il parvient à trouver l'optimum de petites instances d'environ 20 clients. Au delà, l'écart entre les bornes supérieure et inférieure reste grand. Il vaut alors mieux recourir à l'approche heuristique. C'est ce que propose Crainic *et al.* (2008) par une méthode intitulée en anglais fast cluster-based heuristic method. Bien que les résultats n'aient pas permis d'améliorer fortement l'écart aux bornes inférieures, une analyse intéressante est réalisée sur l'aspect de localisation de satellites.

Un autre sous-problème proche, le problème de localisation à deux niveaux ne considère pas de tournées puisque les clients sont servis par un camion dédié. Tragantalerngsak *et al.* (2007) proposent un modèle mathématique et six heuristiques basées sur une relaxation lagrangienne suivie d'une optimisation par sous-gradients pour ce problème. Kaufman *et al.* (1977) et Ro et Tcha (1984) développent des algorithmes de séparation et évaluation. Gao et Robinson Jr. (1992) traitent le cas sans contrainte de capacité aux deux niveaux et appliquent une méthode de type dual-based branch and bound. Plus récemment, Gendron et Semet (2009) étudient un problème de localisation-distribution à trois niveaux de distribution (hubs, dépôts, satellites, clients). Ils proposent un modèle mathématique ainsi que des méthodes de relaxation et des méthodes de recherche à voisinages variables.

Un autre problème relié au LRP-2E est celui des tournées de véhicules multi- dépôts avec des satellites (Multi-Depot Vehicle Routing Problem with Satellite Facilities - MDVRP-SF). Il s'agit d'un VRP multi-dépôts dans lequel les tournées peuvent visiter plus de clients que leurs capacités leur permettent, par le biais d'un rechargement dans un satellite. Ainsi, les véhicules effectuent ce qu'on appelle des rotations, limitée non par une capacité mais par une durée totale de livraison/collecte. Une version simple du MDVRP-SF est présentée par Jordan et Burns (1984). Angelelli et Speranza (2002) développe une heuristique pour un problème de tournées périodiques (PVRP) dans lequel le déchargement à des satellites est autorisé.

Crevier *et al.* (2007) étudient un système de distribution à Montréal et proposent une heuristique combinant une mémoire adaptative et des sous-problèmes résolus par une recherche taboue et par programmation linéaire en nombres entiers.

Récemment, Gonzalez-Feliu (2010) a présenté, entre autres, un modèle très général pour les problèmes de localisation-routage multi-niveaux (N-Echelon Location Routing Problem - LRP-NE).

L'étude menée sur le problème de localisation-routage à deux niveaux a abouti au développement de méthodes heuristiques visant à résoudre des instances ayant jusqu'à 200 clients et de 5 à 20 satellites. Plus précisément, il s'agit d'heuristiques constructives, d'une métaheuristique de type GRASP, renforcée par une technique d'apprentissage et un processus de path-relinking (PR), d'une métaheuristique à démarrages multiples de type recherche locale itérée et sa version hybride avec un PR, et une troisième métaheuristique basée sur un algorithme mémétique intégrant un recuit simulé, en guise de recherche locale. En outre, une méthode exacte de branchement-et-coupes et des bornes inférieures ont été proposées.

3.2.5 Problèmes multi-périodes

La gestion des stocks est une problématique intéressante et difficile au sein d'une chaîne logistique. En effet, en considérant un horizon de planification multi-période, il est probable que les besoins des clients ne soient pas constants. Après une étude sur les approvisionnements dans un système MRP (Material Requirements Planning) (Dolgui *et al.*, 2005; Dolgui et Prodhon, 2007), la livraison des clients tenant compte de leurs capacités de stockage offre une autre perspective de recherche intégrant de nouvelles contraintes au problème de tournées. Dans les variantes étudiées précédemment, il est considéré que le service des clients correspond exactement à leur demande. Ceci implique qu'ils possèdent un magasin/entrepôt de taille suffisante où déposer cet arrivage. L'idée est maintenant de faire l'hypothèse que leurs espaces de stockage sont limités.

Il en résulte deux déclinaisons :

- le cas périodique, dans lequel les besoins des clients sont réguliers, mais selon leur désir, une fréquence de visite est définie sur un horizon cyclique.
- le cas avec gestion des stocks, dans lequel les besoins sont irréguliers mais déterministes sur un horizon défini, et les clients laissent le soin au transporteur de définir les jours de visite et les quantités servies.

Le premier cas est abordé à travers le problème de localisation-routage. Pour le second, le LRP mais aussi le HVRP sont étudiés.

3.2.5.1 Le problème de localisation-routage périodique

L'étude d'une extension des problèmes de localisation-routage au cas multi-périodes est menée en parallèle des travaux précédemment présentés. Il s'agit du Periodic LRP ou PLRP. Ce problème n'avait a priori jamais été étudié dans la littérature avant 2007.

Pour chaque client une sélection d'une combinaison de jours de service doit être effectuée. L'objectif devient de trouver quel sous-ensemble de dépôts ouvrir, quelles combinaisons de jours de visite affecter aux clients et quelles tournées réaliser, de manière à minimiser l'ensemble des coûts. Ce problème a fait l'objet de collaborations avec Christian Prins. Il en a découlé diverses publications en conférences internationales (Prins et Prodhon, 2008; Prodhon, 2008a,b, 2009a,b,c, 2010) et un article de revue internationale (Prodhon, 2011).

Outre le sous-problème de localisation-routage détaillé en Section 3.2.1, le problème périodique de tournées de véhicules (PVRP), introduit par Christofides et Beasley (1984), a reçu une certaine attention. Il existe des versions de tournées sur arcs (Chu *et al.*, 2006; Lacomme *et al.*, 2005) mais la plupart des publications concernent les tournées sur nœuds. Les méthodes de résolution sont majoritairement heuristiques (Christofides et Beasley, 1984; Tan et Beasley, 1984; Chao *et al.*, 1995). Une approche très performante est proposée par Cordeau *et al.* (1997). Il s'agit d'une méthode de type tabou dédiée au multi-dépôts VRP (MDVRP). En effet, il est possible sous certaines conditions de transformer le PVRP en multi-dépôts VRP et inversement. Pour cela, l'affectation aux périodes est comparée à une affectation à un dépôt. Des résolutions exactes du MD-VRP sont alors disponibles (Laporte *et al.*, 1984, 1988) permettant de résoudre des instances ayant jusqu'à 80 clients (dans le cas asymétrique). Plus récemment, Hemmelmayr *et al.* (2009) développent une recherche à voisinage variable qui obtient en moyenne les meilleurs résultats.

Différentes approches heuristiques sont issues de cette étude sur le PLRP. La première consiste à résoudre les sous-problèmes de localisation de dépôts, d'affectation des clients à des jours de service et de tournées cycliquement mais pas indépendamment puisque des échanges d'informations sont réalisés à chaque itération. De plus, une recherche locale permet des améliorations sur les tournées (donc jour par jour), mais également des modifications d'affectation aux périodes. Un algorithme génétique à démarrage multiple et une recherche locale évolutionnaire considérant plusieurs niveaux de décision simultanément découlent également de cette recherche.

3.2.5.2 Le problème de localisation-routage avec gestion des stocks

Le problème traité ici consiste à concevoir et gérer une chaîne logistique. Il s'agit de combiner le LRP, consistant à déterminer l'ensemble de dépôts à ouvrir pour livrer des clients par le biais d'une flotte de véhicules de capacité limitée, avec une vision multi-période incluant une gestion des stocks chez les clients. Il en résulte le problème d'Inventory LRP - ILRP.

La principale différence avec le PLRP vient du fait que les quantités à livrer ne correspondent plus aux demandes par période, mais consistent en une optimisation minimisant le coût d'équilibre entre les ouvertures de dépôts, le stockage chez les détaillants et la livraison par le biais des tournées. Il faut noter qu'ici, la demande est déterministe mais irrégulière dans le temps, que les dépôts, les clients et les véhicules ont des capacités limitées, et que les clients ne sont livrés qu'à partir d'un seul dépôt tout au long de l'horizon de planification.

L'ILRP n'a pas été beaucoup étudié dans la littérature, en particulier dans la version proposée ici. On peut cependant citer Liu et Lee (2003); Liu et Lin (2005); Ahmadi Javid et Azad (2010) et les présentations en conférences de Wang *et al.* (2008); Zhang *et al.* (2008); Xuefeng (2010).

Par contre, un sous-problème assez proche a reçu plus d'attention, en particulier depuis l'application du système *vendor-managed inventory* (VMI) : l'Inventory-Routing Problem (IRP) dans lequel les décisions de localisation sont fixées (Çetinkaya et Lee, 2000; Campbell *et al.*, 2001; Archetti *et al.*, 2007; Zhao *et al.*, 2010). Un état de l'art a récemment été établi par Andersson *et al.* (2010) sur les modèles statiques et dynamiques. Il est également essentiel de s'intéresser aux travaux sur les chaînes logistiques, même si souvent les livraisons sont considérées directes. Kébé *et al.* (2012) par exemple étudient l'approvisionnement sur deux échelons (fournisseurs, entrepôt, usine) avec une capacité de stockage limitée à l'usine (destinataire) et entre la plate-forme de distribution et l'usine (transport). Enfin, des questions d'entreposage étant en cause et l'espace limité, il serait judicieux de coupler cette recherche avec des études de bin-packing multi-dimensionnel (El Hayek *et al.*, 2008).

Enfin, lorsque les coûts des produits sont reliés au dépôt, on peut se rapprocher du problème de production-distribution intégrées (Integrated Production-Distribution Problem - IPDP). Boudia et Prins (2009) proposent un algorithme mémétique avec gestion de la population (MA|PM) pour ce problème.

L'ILRP est au cœur de la thèse en cours de William Guerrero réalisée en cotutelle avec une équipe de l'université des Andes en Colombie. Il en a découlé des présentations en conférences (Guerrero *et al.*, 2011a,b, 2012a,b).

Une approche de résolution proposée consiste en une heuristique coopérative qui résout les sous-problèmes de localisation-affectation, gestion des stocks et routage par alternance tout en échangeant de l'information afin d'obtenir une optimisation globale du système. Cette approche, comparée à une résolution séquentielle, met en évidence l'intérêt d'une optimisation intégrant les divers niveaux de décision.

3.2.5.3 Le problème de tournées de véhicules à flotte hétérogène avec gestion des stocks

Enfin dans les extensions multi-périodes des problèmes de tournées, une étude émergente porte sur le cas avec flotte hétérogène incluant une gestion des stocks chez les clients. Comme pour l'ILRP, la particularité tient dans le fait que la demande n'est pas régulière dans le temps et que l'optimisation du problème complet revient à trouver un équilibre entre coûts de stockage et coût de routage, tout en respectant ici les capacités et la flotte limitée et hétérogène de véhicules. Ce problème fait l'objet de travaux en collaboration avec Christophe Duhamel et Philippe Lacomme du LIMOS à Clermont-Ferrand, présentés en conférence (Duhamel *et al.*, 2012b).

Pour résoudre ce problème, une approche heuristique hybride est en développement. Elle est basée sur un algorithme de recherche locale évolutionnaire et la stratégie Split qui découpe une séquence de clients en un ensemble de tournées réalisables. Les caractéristiques spécifiques du problème induisent adaptation de résolution en particulier dans la génération de solutions et l'exploration de l'espace des solutions.

3.2.6 Problèmes multi-critères

En dehors des recherches sur des problématiques relatives à la logistique du transport, je m'intéresse à d'autres pistes faisant appel à de l'optimisation, en particulier l'optimisation multi-objectif. L'idée est de maximiser (ou minimiser selon les cas) plusieurs critères à la fois tout en respectant un certain nombre de contraintes caractérisant le problème modélisé.

3.2.6.1 Set-Covering Problem

Le premier problème multi-objectif étudié concerne un problème d'optimisation combinatoire, le problème de recouvrement (le Set Covering Problem - SCP).

L'optimisation combinatoire est extrêmement performante pour résoudre des problèmes réels de décision. Cependant, la modélisation se limite souvent à la considération d'un seul critère, alors que les décisionnaires font en réalité face à divers objectifs souvent conflictuels. Ehrgott et Gandibleux (2000) montrent d'ailleurs que très peu de publications sur l'optimisation combinatoire multi-critère (MOCO) ont été réalisées avant 1990.

Contrairement à l'optimisation mono-objectif, pour un problème multi-objectif (P) il n'existe pas de solution optimale à proprement parlé mais un ensemble de solutions non dominées ou efficaces, aussi appelé ensemble des solutions Pareto Optimales. Pour chacune de ces solutions, aucune amélioration ne peut être faite sur l'un des objectifs sans détériorer la valeur sur un des autres objectifs. Ces solutions forment le front de Pareto.

Plusieurs solutions, différentes dans l'espace des décisions, peuvent être équivalentes en termes d'objectifs, c'est-à-dire avoir les mêmes valeurs pour chacun des critères à optimiser. La recherche de solutions diversifiées (côté objectifs) mène à la détermination de l'ensemble complet minimum $E(P)$. Ceci est particulièrement adapté en cas de problème comportant suffisamment de solutions pour se satisfaire d'une telle approximation de l'ensemble non dominé. Sinon, il est intéressant de rechercher l'ensemble maximal complet comprenant également les solutions de même valeur.

En outre, deux types de solutions existent sur le front Pareto : l'ensemble des solutions dites supportées $SE(P)$ et celles dites non supportées $NSE(P)$. Les premières sont celles pouvant être retrouvées en résolvant la version mono-objectif des combinaisons linéaires convexes des différentes fonctions-objectifs. Seulement, en optimisation combinatoire, souvent $E(P) \neq SE(P)$ et cette approche n'est pas suffisante.

Pour le SCP, nous considérons le cas bi-objectif (BOSCP), qui est NP-difficile.

Le SCP se définit ainsi. Soit $A = (a_{ij})$ une matrice binaire de dimension $m \times n$, et les ensembles $M = \{1, \ldots, m\}$ et $N = \{1, \ldots n\}$. Pour chaque $j \in N$ (colonne de A) est associé un coût c_j. Une colonne $j \in N$ couvre une ligne $i \in M$ si $a_{ij} = 1$. Un sous-ensemble $S \subseteq N$ est appelé une couverture si chaque ligne $i \in M$ est couverte par au moins une colonne $j \in S$. Le coût associé à S est défini par $\sum_{j \in S} c_j$. Le but est de trouver le sous-ensemble S minimisant le coût de couverture.

Une approche en deux-phases a été fructueusement appliquée à plusieurs problèmes de type MOCO (Tuyttens *et al.*, 2000; Visee *et al.*, 1998), mais pas au MOSCP au moment de l'étude. L'idée consiste d'abord à chercher les solutions supportées $SE(P)$ en utilisant une technique de graduation pour générer et résoudre des problèmes mono-objectifs. La seconde phase recherche des solutions potentiellement non-supportées $NSE(P)$ par des méthodes spécifiques au problème.

La proposition faite ici consiste à résoudre la première phase en utilisant une heuristique efficace pour le SCP, basée sur une relaxation lagrangienne avec optimisation des sous-gradients (Cordone *et al.*, 2001). Pour la seconde, une autre résolution heuristique, s'appuyant sur les résultats obtenus durant la relaxation lagrangienne (coûts réduits, bornes), recherche successivement des solutions non supportées entre chaque paire adjacente de solutions de $SE(P)$ (voir Prins *et al.* (2006d)).

3.2.6.2 Plasmonique

Parmi les applications possibles de l'optimisation multi-objectif, une est particulièrement prometteuse et fait l'objet d'une étude réalisée conjointement entre le Laboratoire de Nanotechnologie et d'Instrumentation Optique (LNIO) et le Laboratoire d'Optimisation des Système industriel (LOSI). Il s'agit des nanotechnologies et plus précisément ce qui concerne le développement de capteurs pour la santé, la détection de composants chimiques en très faibles quantités ... En effet, pour développer des capteurs efficaces, il est nécessaire d'optimiser les nanostructures. Cependant, il n'est pas possible ou trop coûteux de fabriquer de nombreux prototypes. Il est donc indispensable de passer par la voie de la modélisation.

L'idée de ce projet est de réaliser une modélisation permettant de décrire correctement les propriétés des nanostructures, leurs interactions avec la lumière et les composés à détecter afin de l'intégrer dans des méthodes dites *inverses* (gradients conjugués, recuit simulé, méthodes évolutionnaires) pour déterminer les paramètres optimaux recherchés des nanostructures. Les premiers résultats de ces travaux ont fait l'objet de deux conférences internationales (Prodhon *et al.*, 2009a,b).

Le principal constat lors des résultats expérimentaux est que la maximisation de la visibilité ou du contraste des interférences par l'optimisation de la surface des supports plasmoniques n'apporte pas de solution idéale dans laquelle les deux critères sont maximisés. L'idée est alors de discuter de l'influence de la fonction-objectif dans le contexte de la lithographie plasmonique et de proposer une étude simultanée des deux objectifs dans un algorithme de type NSGA2.

3.3 Approches de résolution

Dans la partie précédente, l'accent a porté sur les thématiques de recherche qui ont abouti à diverses publications. L'essentiel porte sur la recherche opérationnelle pour des problèmes d'optimisation combinatoire de transport, qui comportent en sous-problème celui des tournées de véhicules, le VRP. Ils sont donc plus complexes que le VRP, lui-même plus complexe que le TSP qui est déjà *NP*-difficile.

La résolution de tels problèmes par des approches exactes est souvent exposée à des problèmes de mémoire ou de temps de calcul trop longs. C'est pourquoi aujourd'hui, seules des petites instances (une centaine de clients pour le VRP, une cinquantaine au plus pour le LRP par exemple) peuvent être résolues optimalement. Ainsi, ce sont principalement les méthodes heuristiques qui sont développées.

D'une manière générale, les problèmes étudiés sont d'abord modélisés sous forme d'un programme mathématique. Ensuite, une méthode de résolution est envisagée selon l'analyse des propriétés observées.

3.3.1 Résolutions exactes

Un problème d'optimisation combinatoire se formule souvent sous la forme d'un programme linéaire en nombres entiers (*PLNE*), dans lequel une fonction-objectif linéaire, soumise à un ensemble de contraintes linéaires, est à optimiser. D'un point de vue géométrique, ces contraintes forment un polyèdre (ensemble de solutions d'un système fini d'inégalités linéaires).

De tels problèmes ne sont pas abordables par les outils de l'analyse mathématique, à moins de les approximer en utilisant des variables à valeurs réelles. La relaxation continue d'un *PLNE* peut se résoudre par l'algorithme du simplexe par exemple. Malheureusement, la solution optimale entière peut être très éloignée de la relaxation continue. En nombres entiers, il faut avoir recours à des méthodes telles que les algorithmes de *séparation et évaluation* ou les *méthodes de coupes*. Ces dernières, relativement performantes pour les problèmes de tournées, sont détaillées ci-après.

3.3.1.1 Approche polyédrale

Les algorithmes de coupes sont une approche de résolution de type polyédrale. Ce genre de méthode apporte généralement de bons résultats sur les problèmes de tournées (voir par exemple les articles de Padberg et Rinaldi (1991) pour le *TSP* et de Augerat *et al.* (1999) ou Lysgaard *et al.* (2004) pour le *VRP*).

Les itérations d'un algorithme de coupes consistent à résoudre un programme linéaire. Au départ, certaines contraintes du problème traité sont relaxées. Ensuite, pour chaque solution optimale du problème obtenu, on recherche une ou plusieurs contraintes violées pour le problème d'origine. Elles sont alors ajoutées au programme linéaire à résoudre. Pour un problème de minimisation par exemple, ce procédé se poursuit soit jusqu'à l'obtention d'une solution réalisable ou de coût égal à une borne supérieure, soit jusqu'a ce que l'on ne trouve plus de contraintes violées. Ce dernier cas fournit une borne inférieure. Si cette borne est calculée à chaque nœud d'un algorithme de séparation et évaluation, la méthode s'appelle *branchement et coupes*.

3.3.1.2 Applications

Le développement de méthodes exactes est ici réalisé en priorité aux problèmes nouveaux dans la littérature, au moins dans leur formulation. Il s'agit d'abord des problèmes de localisation-routage avec contraintes de capacité (CLRP), puis des versions à deux niveaux (STTRPSD et LRP-2E).

L'idée est de proposer de nouvelles bornes inférieures. Pour cela, il est nécessaire de se baser sur des formulations mathématiques appropriées aux méthodes visées, les algorithmes de coupes. Ainsi, les programmes linéaires générés utilisent des variables à 2 indices, permettant de réduire le nombre de variables de décision par rapport aux modèles à 3 indices par exemple.

Voici à titre d'exemple un modèle développé pour le CLRP.

Les variables de décision suivantes sont utilisées : $y_i = 1$ si et seulement si le dépôt i est ouvert, $x_{ij} = 1$ $(i, j \in V)$ si et seulement si un véhicule utilise l'arête (i, j) une et une seule fois, $x_{ij} = 2$ $(j \in J,\ i \in I)$ si et seulement si un véhicule utilise une même arête (i, j) deux fois (si un véhicule ne sert qu'un seul client j à partir du dépôt i). Les variables x_{ij} avec $(i, j) \in I$ peuvent être exclues.

Les données du problème sont les ensembles I des m dépôts disponibles et J des n clients à visiter, et la distance c_{ij} entre chacun d'entre-eux. La demande d_j du client $j \in J$ est connue. Un dépôt $i \in I$ a un coût d'ouverture O_i et une capacité limitée W_i. La visite des clients se fait par des véhicules de capacité Q engendrant un coût fixe d'utilisation F.

Mais avant d'exposer le modèle, les notations classiques suivantes sont rappelées :

- $\forall H \subseteq E, \qquad x(H) = \sum_{(i,j)\in H} x_{ij}$;
- $\forall S \subseteq J, \qquad D(S) = \sum_{j\in S} d_j$;
- $\forall S \subseteq V, \qquad \delta(S)$ correspond à l'ensemble des arêtes avec une extrémité dans S et l'autre dans l'ensemble $V\backslash S$;
- $\forall S \subseteq V, \qquad \gamma(S)$ représente l'ensemble des arêtes ayant leurs deux extrémités dans S;
- $\forall S \subseteq V$ et $\forall S' \subseteq V\backslash S, \qquad E(S : S')$ correspond à l'ensemble des arêtes avec une extrémité dans S et l'autre dans S'.

$$\text{(LRP2)} \quad \min \sum_{(ij)\in E} c_{ij}x_{ij} + \frac{F}{2}\sum_{i\in I}\sum_{j\in J} x_{ij} + \sum_{i\in I} O_i y_i \tag{3.1}$$

sous les contraintes :

$$x(\delta(j)) = 2 \qquad \forall j \in J \tag{3.2}$$

$$x(\delta(S)) \geq 2\lceil D(S)/Q\rceil \qquad \forall S \subseteq J \tag{3.3}$$

$$x_{ij} \leq 2y_i \qquad \forall i \in I, \quad \forall j \in J \tag{3.4}$$

$$x(\delta(S \cup \{i\})) \geq 2 \qquad \forall i \in I, \quad \forall S \subseteq J \quad | \quad D(S) > W_i \tag{3.5}$$

$$2x(\gamma(S\cup\{j\}))+x_{ij}-x(E(S:((J\backslash(S\cup\{j\})))\cup\{i\})) \leq 2|S|$$

$$S \subseteq J\backslash\{j\} \qquad \forall i \in I \qquad S \neq \emptyset \tag{3.6}$$

$$x_{ij} \in \{0,1,2\} \qquad \forall i \in I, \quad \forall j \in J \tag{3.7}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,j \in E \tag{3.8}$$

$$y_i \in \{0,1\} \qquad \forall i \in I \tag{3.9}$$

Dans ce programme linéaire, la fonction-objectif (3.1) minimise les coûts d'ouverture de tournées et de dépôts ainsi que les coûts pour visiter les clients. Les contraintes (3.2) concernent le degré des nœuds dans les tournées. Les contraintes (3.3) correspondent au respect des capacités des véhicules. En effet, le terme $\lceil D(S)/Q\rceil$ correspond au nombre minimum de véhicules nécessaires pour couvrir la demande des clients de l'ensemble S. Comme au moins deux arêtes sortent d'un ensemble de clients contenu dans une même tournée afin de la relier à un dépôt, le membre de gauche de l'inégalité doit être au moins égal au double du nombre de tournées minimum nécessaires pour servir l'ensemble S. Les contraintes (3.4) imposent que les arêtes ne soient reliées aux dépôts que si ces derniers sont ouverts. Les contraintes (3.5), dans le même esprit que les contraintes (3.3), obligent à respecter les capacités des dépôts. (3.6) sont une nouvelle formulation de contraintes généralisant les propositions faites par Laporte *et al.* (1986). Elles restreignent l'affectation d'un véhicule à un seul dépôt et sont appelées contraintes *path-f1* car elles interdisent ainsi les chemins entre deux dépôts distincts. (3.7 - 3.9) sont les contraintes d'intégrité des variables.

Afin d'améliorer les bornes issues de la littérature, de nouvelles familles de contraintes valides sont développées, ainsi que des procédures dites de séparation pour identifier les contraintes non respectées dans une solution du problème relaxé.

Par exemple, des contraintes de type *y-capacité* sont utilisées pour la formulation que nous venons de donner du CLRP. Soit $k(S) = \lceil D(S)/Q\rceil$, représentant le nombre minimum de véhicules nécessaires pour servir S, et I_j les dépôts auxquels est relié le client j. On a alors :

$$\sum_{j \in S'} \left(\sum_{i \in I_j} (x_{ij} - y_i) \right) + x(\gamma(S)) \leq |\, S \,| - 1 \qquad S \subseteq J,\ S' \subset S \ /\ |\, S' \,| \leq |\, S \,| - 1,$$
$$k(S) = 1,\ I_j \subseteq I \text{ pour } j \in S' \quad (3.10)$$

Ces contraintes peuvent se généraliser pour toute valeur $k(S) \geq 1$ de la manière suivante :

$$\sum_{j \in S'} \left(\sum_{i \in I_j} (x_{ij} - y_i) \right) + x(\gamma(S)) \leq |\, S \,| - k(S) \qquad S \subseteq J,\ S' \subseteq S \ /\ |\, S' \,| \leq |\, S \,| - k(S),$$
$$k(S \backslash S') = k(S),\ I_j \subseteq I \text{ pour } j \in S' \quad (3.11)$$

Elles caractérisent le fait qu'un dépôt ne peut être ouvert partiellement.

On peut aussi citer des contraintes alternatives aux inégalités appelées *chain barring* proposées par Laporte *et al.* (1986) :

$$\sum_{i \in I'} x_{ij} + 2x(\gamma(S \cup \{j, l\})) + \sum_{k \in I \backslash I'} x_{kl} \leq 2|S| + 3$$
$$j, l \in J, \quad S \subseteq J \backslash \{j, l\}, \quad S \neq \emptyset, \quad D(S) \leq Q, \quad I' \subset I \quad (3.12)$$

et si $S = \emptyset$:

$$\sum_{i \in I'} x_{ij} + 3x_{lj} + \sum_{k \in I \backslash I'} x_{kl} \leq 4 \quad j, l \in J, \quad I' \subset I \quad (3.13)$$

Ces dernières ont pour caractéristique de travailler non pas sur le fait qu'une tournée doive être affectée à un unique dépôt mais, de manière plus générale, à un unique sous-ensemble de dépôts.

3.3.1.3 Algorithmes

Une résolution exacte basée sur l'approche polyédrale est proposée pour des problèmes combinant à la fois un aspect localisation et un aspect routage. Trois types de résultats sont alors envisageables :

- Algorithme de coupes (résultat à la racine de l'arbre de branchement)
- Algorithme de branchement et coupes partiel (branchement uniquement sur les variables d'ouverture des dépôts/satellites)
- Algorithme de branchement et coupes complet

Des bornes inférieures pour des instances de petites à grande taille sont ainsi obtenues, mais également des solutions optimales pour des instances allant jusqu'à 50 clients et 5 dépôts pour le CLRP, 100 clients et 20 dépôts pour le STTRPSD et quelques instances à 50 clients et 10 dépôts pour le LRP-2E.

Outre les algorithmes de coupes, des modèles sont aussi formulés et testés pour le LRP avec gestion des stocks.

Ces travaux ont fait l'objet de diverses publications (Belenguer *et al.*, 2011; Nguyen *et al.*, 2011a; Guerrero *et al.*, 2011b; Villegas *et al.*, 2010a) dont une partie, sur les méthodes polyédrales, a été faite en collaboration avec José-Manuel Belenguer et Enrique Benavent de l'université de Valencia, Espagne :

> J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. Wolfler-Calvo. A Branch-and-Cut method for the Capacitated Location-Routing Problem. *Computers & Operations Research*, 38, pp. 931-941, 2011.
>
> J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. Wolfler-Calvo. A Branch-and-Cut method for the Capacitated Location-Routing Problem. *Computers & Operations Research*, 38, pp. 931-941, 2011.
>
> V.-P. Nguyen, C. Prins, C. Prodhon. A cutting plane method for the Two-Echelon Location Routing Problem. *Optimization 2011*, Lisbon, Portugal, 2011.
>
> W.J. Guerrero, N. Velasco, C.A. Amaya , C. Prodhon. The Inventory Location Routing Problem with Deterministic Demand. *POMS Annual Conference*, 2011.
>
> J.G. Villegas, J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon. A cutting plane approach for the single truck and trailer routing problem with satellite depots. *EURO 2010* (XXIV European Conference On Operational Research), Lisbon, Portugal, July 11-14, 2010.

3.3.2 Heuristiques

Les algorithmes exacts ne permettant pas une résolution des problèmes traités de grande taille, le recours aux méthodes approchées s'impose. Des heuristiques et recherches locales sont d'abord présentées, elles sont à la base des méthodes de résolution plus sophistiquées détaillées ensuite, telles que les métaheuristiques.

3.3.2.1 Heuristiques constructives

Les heuristiques sont des méthodes de résolution approchées, souvent basées sur le bon sens ou sur des observations empiriques, qui permettent d'obtenir des solutions réalisables. La construction d'une telle solution résulte en général de décisions élémentaires consécutives, chaque élément ajouté étant sélectionné de manière gloutonne. Cela signifie que l'attribut choisi parmi les différentes possibilités pour étendre la solution partielle est celui qui optimise un certain critère.

La principale contribution concernant les heuristiques gloutonnes est la généralisation de l'algorithme de Clarke et Wright intégrant les aspects multi-dépôts et localisation (Extended Clarke & Wright Algorithm - ECWA). Il a été initialement créé pour le LRP.

L'algorithme classique mono-dépôt de Clarke et Wright (1964) (CWA) commence par dédier une tournée à chaque client de façon à le relier à l'unique dépôt. Dans une version multi-dépôts, les clients ne sont plus tous reliés à un site unique. Ils sont affectés au plus proche dépôt ayant encore suffisamment de capacité résiduelle pour les servir. Le résultat donne une solution initiale triviale ayant une allure de *bouquet de marguerites*. Ensuite, le but est de fusionner les tournées obtenues afin de réduire le coût de la solution courante.

Une itération de l'ECWA examine toutes les fusions possibles et réalise celle qui produit le plus gros gain G. Ainsi, pour chaque paire de tournées, $4m$ fusions sont envisageables (avec m, le nombre de dépôts disponibles). En effet, la tournée résultant d'une fusion peut maintenant être affectée à n'importe quel dépôt disponible de capacité compatible (voir figure 3.1). La complexité de $ECWA$ est $O(mn^3)$.

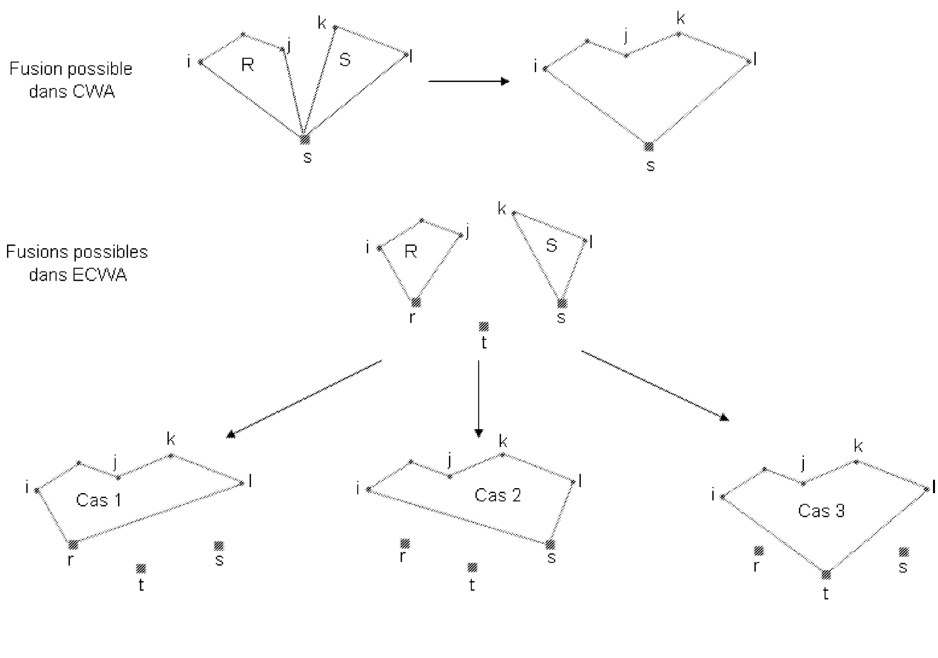

Figure 3.1: Fusions dans CWA et ECWA - Prins et al. *4OR*, 2006

Cet algorithme est utilisé en version randomisée pour générer des solutions au sein d'un GRASP (Greedy Randomized Adaptive Search Procedure) pour le LRP, mais aussi pour le LRP-2E et le PLRP.

La principale référence sur le sujet est Prins *et al.* (2006c) :

> C. Prins, C. Prodhon, R. Wolfler Calvo, Solving the Capacitated Location-Routing Problem by a GRASP complemented by a Learning Process and a Path Relinking, *4OR - A Quarterly Journal of Operations Research*, 4(3), pp. 221-238, 2006.

3.3.2.2 Split

Une stratégie intéressante dans l'exploration des solutions consiste à changer d'espace de recherche. Plus précisément, dans le cadre des problèmes de tournées qui cherchent dans la plupart des cas à déterminer un plus court chemin soumis à certaines contraintes dans le but de servir un ensemble de clients, il est possible d'avoir recours à une représentation indirecte de la solution en générant une séquence reprenant tous les clients (un tour géant ou permutation T de n clients). L'exploration de cet espace, plus restreint que l'original, s'avère astucieux à condition de pouvoir re-basculer dans l'espace initial. Pour cela, il suffit de découper la séquence en tournées tenant compte des contraintes spécifiques au problème original pour obtenir une solution réalisable. Mais il est clair que ce découpage est un élément-clé qui se doit d'être efficace.

La méthode Split, initiée par Beasley (1983) en tant que seconde phase d'une approche *route-first, cluster-second* pour le VRP peut répondre à cette attente.

Dans sa version originale, la première phase génère un tour géant visitant tous les clients (par résolution d'un TSP - Traveling Salesman Problem). Puis un découpage de cette séquence, tenant compte des contraintes de capacité et de longueur maximale de tournée, conduit à une solution du problème de tournées. Ce principe est repris en 2004 dans des algorithmes mémétiques pour la résolution du problème de tournées sur arcs (CARP - Capacitated Arc Routing Problem) (Lacomme *et al.*, 2004), meilleure méthode publiée jusqu'en 2008, et pour le VRP (Prins, 2004), également meilleure méthode, récemment détrônée par une autre métaheuristique utilisant Split (Prins, 2009a). Les excellents résultats obtenus ont encouragé le développement d'une structure orientée-objet dédiée au développement rapide d'heuristiques basées sur ce principe. Cet outil fournit un ensemble de composants facilement utilisables pour traiter différentes extensions du VRP (Villegas *et al.*, 2008).

Dans la version de Prins, le découpage du tour géant se fait de manière optimale de la manière suivante. Pour une séquence de p clients, on construit un graphe auxiliaire $H = (X, A, Z)$ avec X un ensemble de $p + 1$ nœuds indicés de 0 à p, et un ensemble A contenant un arc (i, j), $i < j$, si une tournée desservant les clients de T_{i+1} à T_j (inclus) est réalisable en terme de capacité. Le poids z_{ij} de (i, j) est égal au coût de la tournée allant de i à j. Le découpage optimal de la séquence des p clients correspond à un chemin de coût minimum dans H, allant du nœud 0 au nœud p, comme le montre la

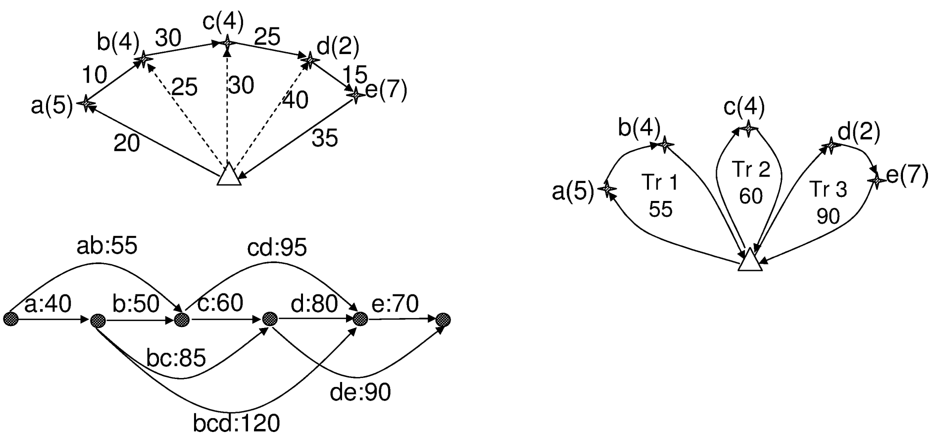

Figure 3.2: Exemple de la procédure Split repris de Prins (2004)

figure 3.2, calculé en temps polynomial par l'algorithme de Bellman par exemple.

Cependant, dans les problèmes qui nous intéressent, des contraintes de ressources supplémentaires sont à prendre en compte. Il faut alors trouver une adaptation adéquate.

La première technique consiste à ne traiter que des sous-problèmes, sans contrainte de ressources. C'est ce qui est proposé initialement pour le LRP. La séquence de clients est pré-découpée, en regroupant dans les sous-séquences les clients par affectation aux dépôts ouverts. Il reste à résoudre un VRP par sous-séquence grâce au Split "basique".

La principale référence pour cette approche est (Prins *et al.*, 2006b) :

> C. Prins, C. Prodhon, R. Wolfler Calvo, A Memetic Algorithm with Population Management (MA|PM) for the Capacitated Location-Routing Problem, dans *Evolutionary computation in combinatorial optimization* (actes d'EvoCOP 2006, avril 2006, Budapest, Hongrie), J. Gottlieb et G.R. Raidl (éd.), Lecture Notes in Computer Science 3906, pp. 183-194, Springer, 2006. ISBN 3-540-33178-6.

Dans la première proposition, la gestion des dépôts est déléguée à une autre partie de l'algorithme. Une autre version, dans laquelle l'affectation des tournées aux dépôts se fait dans Split, est développée. Cependant, il n'y a toujours aucune réelle gestion des ressources puisque les contraintes de capacité des dépôts sont relaxées. La solution résultante n'est pas forcément réalisable et donc une procédure de réparation est nécessaire. Des détails sur cette version sont disponibles dans l'article Nguyen *et al.* (2012b):

> V.-P. Nguyen, C. Prins, C. Prodhon. Solving the Two-Echelon Location Routing Problem by a hybrid GRASP×Path Relinking complemented by a learning process. *European Journal of Operational Research*, 216, pp. 113-126, 2012.

Pour le cas particulier de tournée simple avec remorque (STTRPSD), un découpage optimal pour le problème complet est élaboré tout en gardant un algorithme polynomial. Ici, Split doit fractionner la séquence de clients en tournées de second niveau, mais la difficulté réside dans le fait que la distance entre les satellites doit être prise en compte dans la fonction-objectif. Ainsi, il faut gérer également la sélection et l'ordre de visite des satellites.

Pour cela, le graphe auxiliaire utilisé est composé non pas de $n+1$ nœuds, mais de $n \times p + 2$ nœuds (avec n le nombre de clients et p le nombre de satellites) : deux copies du dépôt principal (a et b), et $n \times p$ nœuds pour les états $[l, j]$ (l représentant un satellite, et j un client). Cependant, les satellites n'ayant pas de contraintes de capacité, on peut difficilement parler d'un cas avec gestion de ressources.

Cette application non-triviale du Split est expliquée dans l'article Villegas *et al.* (2010b) :

> J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5), pp 780-794, 2010.

Pour une réelle gestion de ressources (autre que la capacité des véhicules), il faut calculer un plus court chemin dans le graphe auxiliaire en générant plusieurs labels par nœud (Desrochers, 1988). Une analyse des labels est nécessaire pour ne garder que l'ensemble des non-dominés. Un label sur le nœud i représente une évaluation de la solution partielle considérant les tâches de T_1 à T_i. Il est composé entre-autres d'un vecteur coût (donnant la valeur de la solution partielle pour chaque critère optimisé), d'un vecteur ressource (indiquant les ressources restant encore disponibles à ce niveau de construction de la solution pour chacune des ressources utilisées), d'un couple de valeur définissant le label-père et d'un booléen signalant le statut de propagation de ce label aux enfants.

Dans la version classique du Split, grâce à deux indices i et j, l'algorithme énumère chaque sous-séquence de T_0 à T_{n+1} correspondant à une tournée réalisable et calcule son coût et sa consommation de ressources. Un label initial sur le nœud 0 de coût nul et ayant la totalité des ressources disponibles se propage sur tous les nœuds atteignables. Ensuite l'algorithme passe au nœud suivant et répète le même processus de propagation à partir des labels présents sur ce nœud. Il s'arrête quand le nœud de propagation correspond au nœud final (ou l'atteinte d'un nombre maximal de labels).

Dans Duhamel *et al.* (2011), une autre exploration est suggérée. Il s'agit de visiter les nœuds en profondeur d'abord. Le principe de propagation est le même que précédemment et démarre avec le même label sur le nœud 0. La différence vient de l'avancement dans la séquence. Le but est de marquer le nœud final d'un label le plus rapidement possible. Ainsi, chaque label généré vient s'ajouter sur une pile et, à la fin d'une propagation à partir d'un label père, le prochain considéré est celui en haut de cette pile. L'algorithme s'arrête quand la pile est vide (ou si un nombre maximal de labels est atteint).

Ce principe est appliqué à deux problèmes avec contraintes de ressources : le LRP et le HVRP (Duhamel *et al.*, 2010b, 2011, 2012a, à paraître). L'article dédié à cette nouvelle implémentation du Split au sein d'une métaheuristique est le suivant :

> C. Duhamel, P. Lacomme, C. Prodhon. Efficient frameworks for greedy split and new depth first search split procedures for routing problems. *Computers & Operations Research*, 38, pp. 723-739, 2011.

3.3.2.3 Recherches locales

Les solutions obtenues heuristiquement peuvent être améliorées par des recherches locales. Il s'agit alors de regarder dans un voisinage de la solution courante s'il n'est pas possible d'améliorer cette dernière. Pour ne pas tomber dans l'énumération complète des solutions possibles, le voisinage doit être de taille relativement restreinte. Le voisinage d'une solution courante S est le plus souvent défini implicitement comme l'ensemble des solutions que l'on peut obtenir en appliquant à S une transformation simple appelée mouvement. L'idée est de choisir un mouvement améliorant et à partir de ce nouveau voisin généré, explorer à nouveau le voisinage. La solution obtenue après de telles améliorations successives sera alors un minimum local.

Les mouvements classiques de recherche locale pour les problèmes de tournées consistent par exemple à déplacer un client, à échanger la position de deux clients ou à enlever k arêtes d'un cycle et à reconnecter les chaînes obtenues avec k autres arêtes (k-opt (Lin et Kernighan, 1973)).

Ces mouvements sont ici adaptés pour les problèmes avec plusieurs dépôts et des gestions de ressources (capacité, nombre de véhicules). D'autres mouvements plus sophistiqués sont également proposés afin de modifier les statuts d'ouverture des dépôts (Prins *et al.*, 2004; Nguyen *et al.*, 2012b) :

> C. Prins, C. Prodhon, R. Wolfler Calvo, Nouveaux Algorithmes pour le Problème de Localisation et Routage sous Contraintes de Capacité, *MOSIM'04 (4ème Conférence Francophone de Modélisation et Simulation*, Nantes, France, 01-03/09/2004), pp. 1115-1122, Editions Tec-Doc Lavoisier, 2004. ISBN 2-7430-0731-1.
>
> V.-P. Nguyen, C. Prins, C. Prodhon. Solving the Two-Echelon Location Routing Problem by a hybrid GRASPxPath Relinking complemented by a learning process. *European Journal of Operational Research*, 216, pp. 113-126, 2012.

Pour l'exploration des voisinages, plusieurs stratégies sont possibles. Outre le choix classique entre premier mouvement améliorant et meilleur mouvement, deux types d'implémentation sont envisagés : i) en imbrication (ou union de voisinages) où pour chaque client (ou paire de clients), tous les mouvements proposés sont testés successivement, ii) en descente à voisinage variable où l'ensemble des mouvements d'un voisinage est exploré jusqu'à l'obtention d'un mouvement améliorant, si aucun n'est trouvé, le voisinage suivant est exploré, sinon, la recherche repart sur le premier voisinage.

L'implémentation en imbrication est appliquée dans l'ensemble des articles traitant du CLRP et HVRP. L'implémentation en descente à voisinage variable est utilisée dans les articles sur les problèmes de tournées avec remorques et le LRP-2E.

3.3.3 Métaheuristiques

Souvent plus performantes que des heuristiques couplées à des recherches locales, les métaheuristiques emploient des méthodes visant à éviter les minimums locaux. Après un bref rappel des méthodes de base, les versions plus performantes, développées pour les problèmes étudiés, sont présentées, à savoir des approches hybrides et/ou des matheuristiques.

3.3.3.1 Versions de base

Les métaheuristiques les plus classiques sont celles fondées sur l'exploration d'un voisinage, et forment la première famille. L'algorithme part d'une solution et la fait évoluer à chaque itération sur l'espace de recherche. Les méthodes les plus connues dans cette catégorie sont la méthode *GRASP* (*Greedy Randomized Adaptive Search Procedure*), le recuit simulé (*Simulated Annealing, SA*), la recherche locale itérative (*Iterative Local Search, ILS*) ou encore la recherche taboue (*Tabu Search, TS*).

Une autre approche possible est basée sur l'utilisation d'une population de solutions et constitue la seconde famille. A chaque itération, la métaheuristique fait évoluer un ensemble de solutions en parallèle. Les algorithmes génétiques (*Genetic Algorithms, GA*), de recherche dispersée (*Scatter Search, SS*) ou les algorithmes de colonies de fourmis (*Ant Colony, AC*) sont des exemples courants de ce type de méthodes. Dans le même esprit, on retrouve aussi l'algorithme des chemins reliants (*Path Relinking*) qui est rarement utilisé seul, mais plutôt en complément des autres méthodes.

Une dernière catégorie, à mi-chemin entre les explorations de voisinage et les évolutions de population, contient les recherches locales évolutionnaires (*Evolutionary Local Search, ELS*) qui proposent un bon compromis. Basées sur le principe de la recherche locale itérative, chaque itération ne fait évoluer qu'une seule solution dans l'espace de recherche par le biais de mutations suivies de recherches locales. Mais contrairement à l'ILS, à partir de cette solution courante, ce n'est pas une unique solution qui est générée, mais un ensemble de m solutions. La méthode s'apparente ainsi à une méthode évolutionniste du type $(1/1+m)-ES$.

3.3.3.2 Métaheuristiques hybrides

Les métaheuristiques classiques, bien que largement plus efficaces que les heuristiques constructives, sont souvent peu compétitives devant les dernières générations de méthodes hybrides.

Dans les travaux développés, des versions hybridant divers composants sont proposées. La première hybridation est celle combinant une métaheuristique avec une recherche locale. Dans cette catégorie, l'exemple maître est certainement l'algorithme mémétique qui ajoute une recherche locale sur les solutions-enfants générées par un algorithme génétique. Cependant, ce n'est souvent pas suffisant.

Le composant proposé par Sörensen et Sevaux (2006) est donc ajouté. Il consiste à gérer la population (PM - Population Management) de manière à avoir des individus dispersés dans l'espace des solutions. Voici deux références de MA|PM (Prins *et al.*, 2006b; Prins et Prodhon, 2008) :

> C. Prins, C. Prodhon, R. Wolfler Calvo, A Memetic Algorithm with Population Management (MA|PM) for the Capacitated Location-Routing Problem, dans *Evolutionary computation in combinatorial optimization* (actes d'EvoCOP 2006, avril 2006, Budapest, Hongrie), J. Gottlieb et G.R. Raidl (éd.), Lecture Notes in Computer Science 3906, pp. 183-194, Springer, 2006. ISBN 3-540-33178-6.
>
> C. Prodhon, C. Prins. A Memetic Algorithm with Population Management (MA|PM) for the Periodic Location-Routing Problem, dans *Hybrid Metaheuristics* (actes de Hybrid Metaheuristics 2008, Septembre 2008, Málaga, Espagne), M.J. Blesa *et al.* (éd.), Lecture Notes in Computer Science 5296, pp. 43-57, Springer, 2008. ISBN 978-3-540-88438-5.

Une autre proposition consiste à remplacer les recherches locales simples par des composants plus agressifs afin de renforcer l'exploration. La difficulté est alors de ne pas tomber prématurément dans un bassin d'attraction qui stopperait la recherche. C'est pourquoi l'implémentation à démarrages multiples est souvent préférée.

Une première idée simple est de mettre en place une descente à voisinage variable au sein d'un GRASP (Villegas *et al.*, 2010b) :

> J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5), pp 780-794, 2010.

Une autre possibilité en guise de composants plus agressifs est de remplacer les recherches locales de métaheuristiques classiques par d'autres métaheuristiques, comme l'introduction d'un ILS/ELS au sein d'un GRASP (Villegas *et al.*, 2009a; Duhamel *et al.*, 2010a) :

> J.-G. Villegas, A. Medaglia, C. Prins, C. Prodhon, N. Velasco. GRASP/Evolutionary Local Search Hybrids for a Truck and Trailer Routing Problem, *MIC 2009* (*8th Metaheuristics International Conference*), Humbourg, Allemagne, 13-16/07/2008.
>
> C. Duhamel, P. Lacomme, C. Prins, C. Prodhon. A GRASPxELS Approach for the capacitated Location-Routing Problem. *Computers & Operations Research*, 37, pp. 1912-1923, 2010.

Afin de mieux explorer l'espace de recherche, il est également possible d'avoir recours aux chemins reliants, soit en post-optimisation d'une méthode ou en interne, ou les deux. Cette hybridation est souvent faite avec les algorithmes de type GRASP car dans cette dernière approche, chaque itération de la méthode génère une solution indépendante des précédentes et le PR permet d'ajouter un lien entre les optimums locaux et ainsi mieux explorer l'espace des solutions. Cette hybridation est également proposée avec les ILS ou ELS à démarrage multiple (Nguyen *et al.*, 2012a,b; Villegas *et al.*, 2011a; Prins *et al.*, 2006c). En effet, les redémarrages permettent de diversifier la recherche mais créent, comme dans le GRASP des solutions indépendantes. Le PR insère ainsi un lien entre les optimums locaux et peut permettre d'atteindre de meilleures solutions.

> V.-P. Nguyen, C. Prins, C. Prodhon. A multi-start Iterative Local Search with tabu list and Path Relinking for the Two-Echelon Location Routing Problem. *Engineering Applications of Artificial Intelligence*, 25(1), pp. 56-71, 2012.
>
> V.-P. Nguyen, C. Prins, C. Prodhon. Solving the Two-Echelon Location Routing Problem by a hybrid GRASPxPath Relinking complemented by a learning process. *European Journal of Operational Research*, 216, pp. 113-126, 2012.
>
> J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 38(9), pp. 1319-1334, 2011.
>
> C. Prins, C. Prodhon, R. Wolfler Calvo, Solving the Capacitated Location-Routing Problem by a GRASP complemented by a Learning Process and a Path Relinking, *4OR - A Quarterly Journal of Operations Research*, 4(3), pp. 221-238, 2006.

3.3.3.3 Coopération et matheuristiques

Utiliser des métaheuristiques existantes ou faire des hybridations pour améliorer les performances, sont des approches efficaces lorsqu'elles sont bien implémentées. En particulier, une des clés à leur réussite repose sur les heuristiques et recherches locales dédiées qui sont utilisées en leur sein.

Le challenge est donc d'analyser le problème à résoudre et de trouver la bonne approche de résolution. De plus, les problèmes étudiés étant de plus en plus complexes, ils sont difficilement abordable de manière globale et directe. L'approche *classique*, majoritairement adoptée dans le passé pour l'optimisation des systèmes logistiques, est la décomposition des problèmes, selon les niveaux de décision par exemple. L'avantage d'une telle démarche est la simplification apportée. Cependant, de cette conduite résulte bien souvent une perte de l'optimalité globale car l'intégralité du problème n'est pas prise en compte.

Une des approches proposées consiste à simplifier la résolution en restreignant l'espace de recherche. Cependant, le but n'est pas de décomposer pour résoudre indépendamment les sous-problèmes. Au contraire, l'objectif est de parvenir à travailler sur des problèmes restreints tout en gardant une vision sur le problème original. Une alternance judicieuse entre les espaces de recherche est alors au cœur de la résolution.

La **première option** dans le champs des problèmes de tournées est le recours à l'algorithme Split, introduit dans la section 3.3.2.2. En effet, cela permet de travailler sur l'espace des séquences/ordres de clients (plus petit que celui des tournées), aussi appelé espace des *tours géants*. Une tournée est alors générée en relaxant la capacité des véhicules. Puis une décomposition en tournées par l'algorithme Split permet de revenir au problème original et de travailler directement sur les tournées pour les recherches locales par exemple. Une alternance entre les espaces de recherche est possible en retournant dans celui des tours géants par concaténation des tournées (voir figure 3.3).

L'idée est alors d'explorer massivement l'espace restreint des tours géants par une métaheuristique et d'évaluer les solutions dans l'espace des tournées grâce à l'algorithme Split. Ce principe, initié par Prins (2004), a largement montré son efficacité sur les problèmes de tournées, comme l'atteste encore la récente publication de Prins (2009a).

Le recours à un algorithme exact (ici le calcul d'un plus court chemin dans un graphe auxiliaire) au sein d'une métaheuristique peut faire entrer cette méthodologie dans la catégorie des matheuristiques. De plus, toute solution optimale du problème de tournées peut alors être retrouvée sous condition de bon séquençage dans l'espace des tours géants.

Ce type d'alternance est exploité pour le cas de problèmes de tournées avec contraintes de ressources, impliquant parfois cependant une résolution heuristique du problème du plus court chemin dans le graphe auxiliaire. L'ensemble des publications sur ce sujet a déjà été présenté dans la section 3.3.2.2.

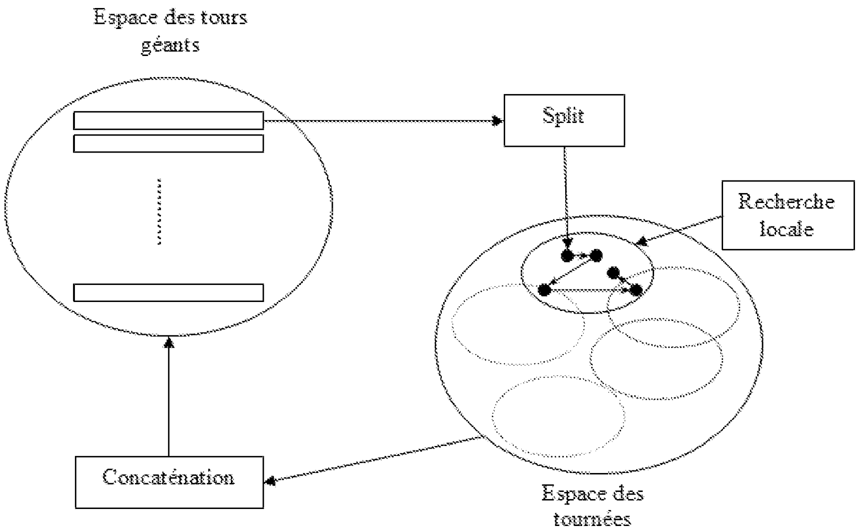

Figure 3.3: Alternance des espaces de recherche avec Split - Duhamel et al. *Computers & Operations Research*, 2010

Une **seconde option** est exploitée pour alterner entre deux espaces de recherche pour le problème de localisation-routage. Cette fois-ci, une première phase résout de manière exacte le problème de localisation et d'affectation par une relaxation Lagrangienne et une optimisation des sous-gradients. La seconde phase résout le problème de tournées de véhicules multi-dépôts par un algorithme tabou. Il s'agit donc encore d'une matheuristique.

Ce n'est pas une approche purement séquentielle qui est proposée puisque ceci n'aboutit généralement pas à une solution globalement optimisée. Le principe repose plutôt sur une alternance entre ces deux phases. Donc après le routage, les dépôts sont "retirer" des tournées, et ces dernières sont converties en *superclients* qui vont être à nouveaux affectés à des dépôts. Une illustration simplifiée de la méthode, appelée LRGTS, est donnée dans la figure 3.4. Cependant, le choix des dépôts et l'affectation des clients ont une forte influence sur les tournées futures, tout comme les tournées déjà construites induisent un choix biaisé de localisation. Il est alors intéressant de réaliser une alternance coopérative qui débute par des *superclients* de petite taille (un seul client) qui grossissent progressivement. Ainsi, la solution évolue en ajustant les différents types de décisions au fil des alternances entre problèmes. Des détails sur cette méthode sont donnés dans l'article de Prins *et al.* (2007) :

> C. Prins, C. Prodhon, P. Soriano, A. Ruiz, R. Wolfler Calvo, Solving the Capacitated Location-Routing Problem by a Cooperative Lagrangean Relaxation-Granular Tabu Search Heuristic, *Transportation Science*, 41(4), pp. 470-483, 2007.

Outre l'alternance entre espaces de recherche, la proposition d'intégration d'une

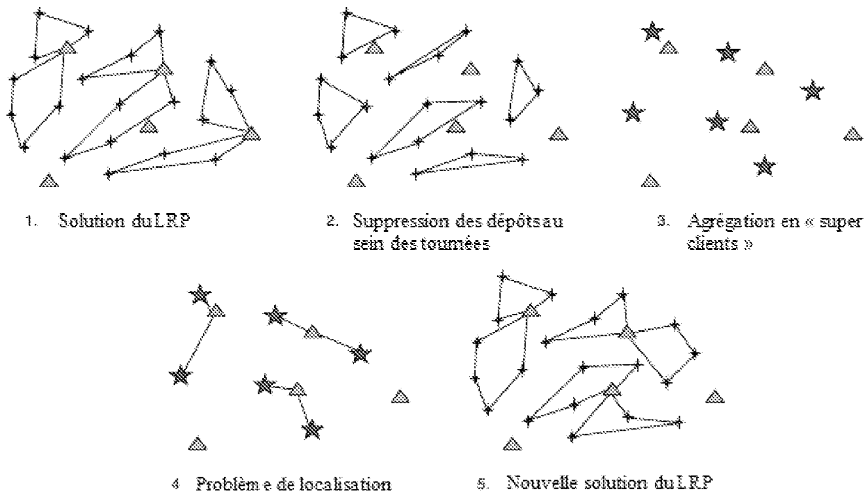

Figure 3.4: Alternance des espaces de recherche avec LRGTS - Prins et al. *Transportation Science*, 2007

résolution exacte, en post-optimisation cette fois-ci, est testée. L'idée est basée sur la génération de colonnes (Feillet, 2010). Dans cette approche, classiquement, le problème est modélisé sous forme d'un problème de partitionnement, dans lequel un sous-ensemble de tournées, parmi celles réalisables, est sélectionné afin de visiter l'ensemble des clients au moindre coût. La difficulté vient dans le fait que le nombre total de tournées réalisables est exponentiel. La résolution proposée consiste à générer un sous-ensemble prometteur par l'intermédiaire d'une métaheuristique qui traite le problème complet, ici un GRASP. A chaque itération, une solution optimale localement est générée pour le problème d'origine et ses tournées sont stockées. A la fin du GRASP, un ensemble de tournées de *bonne* qualité est disponible et le problème formulé sous forme d'un problème de partitionnement est optimalement résolu (sous réserve de l'ensemble de tournées disponibles). Cette matheurisique très efficace est en révision pour une revue internationale (Villegas *et al.*, 2011b, en révision).

> J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. A matheuristic for the Truck and Trailer Routing Problem. *European Journal of Operational Research*, en révision.
>
> J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco Heuristic column generation for the truck and trailer routing problem. *IESM 2011*, (International Conference on Industrial Engineering and Systems Management, Metz, 25-27/05/2011).

Enfin, pour les problèmes de tournées avec gestion de stocks, les travaux en cours combinent une résolution exacte des quantités à livrer par période afin de fournir des indications pour la résolution du problème complet (Guerrero *et al.*, 2011a).

> W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya. A Matheuristic for the Inventory Location Routing Problem with Deterministic Demand. *MIC 2011* (Metaheuristics International Conference), Udine, Italie, 2011.

3.3.4 Approche parallèle

Les méthodes développées peuvent toutes bénéficier de forces du calcul distribuées sur différents processeurs. En effet, depuis quelques années, les résolutions dites parallèles se développent grâce à des grilles de calculs se comportant comme des ordinateurs reliés en parallèle. L'intérêt est essentiellement l'accélération de résolutions lourdes mais ce n'est cependant pas ce qui est visé dans les approches élaborées ici. L'idée est plutôt de travailler de manière coopérative entre les processeurs afin d'améliorer les solutions obtenues. Pour cela, au lieu d'avoir recours à l'acquisition de matériel coûteux, le principe est de tirer parti des modèles d'ordinateurs vendus aujourd'hui et qui possèdent tous des processeurs multi-cœurs (machines équipées de 2 à 8 processeurs partageant une mémoire commune).

Le principe de résolution proposé repose sur une stratégie d'exploration de l'espace des solutions basée sur des recherches locales évolutionnaires (ELS) lancées en parallèle et coopérant au cours des itérations. Les données du problème sont donc centralisées et utilisées par tous les processeurs. Une solution initiale est générée et distribuée à chaque processeur sur lequel est lancé une ELS. Notre application se faisant sur un problème de tournées, la procédure Split est utilisée afin d'alterner entre différents espaces de recherche. Ensuite, une mémoire partagée entre tous les threads stocke les solutions générées pour éviter les visites successives sur les mêmes régions de l'espace de recherche. Une synchronisation entre les processeurs extrait la/les meilleures solutions trouvées afin de les distribuer sur les threads en guise de nouveaux points de départ pour les ELS. Si aucune amélioration n'est observée, la méthode repart d'une nouvelle solution randomisée.

Cette approche fait l'objet de plusieurs communications en conférence (Duhamel *et al.*, 2010c; Lacomme et Prodhon, 2010) et d'un chapitre de livre (Duhamel *et al.*, à paraître).

> C. Duhamel, C. Gouinaud, P. Lacomme, C. Prodhon. A multi-thread GRASPx-ELS for the heterogeneous capacitated vehicle routing problem, dans *Hybrid Metaheuristics*, Springer, en révision.
>
> C. Duhamel, P. Lacomme, C. Prodhon, C. Prins. Parallel Cooperative GRASP for the Heterogeneous Vehicle Routing Problem. *EURO 2010* (XXIV European Conference On Operational Research), Lisbon, Portugal, July 11-14, 2010.
>
> P. Lacomme, C. Prodhon. Parallel Cooperative ELS for the Heterogeneous Vehicle Routing Problem. *EU/MEeting 2010*, Lorient, France, 03-04/06/2010.

3.3.5 Résultats et Conclusion

Les méthodes élaborées pour l'ensemble des problèmes étudiés sont implémentées informatiquement et testées sur des instances classiques de la littérature. Dans quelques cas, certains jeux de test ont été créés. Il s'agit soit de problèmes pour lesquels aucune instance représentant spécifiquement l'étude n'existait, soit de problèmes déjà étudiés mais pour lesquels les ensembles d'essais ne semblaient pas suffisants.

Le but ici est de présenter comment l'évaluation des méthodes proposées est réalisée et de montrer des exemples de résultats.

3.3.5.1 Réglage des paramètres

Une méthode de résolution comprend souvent divers composants. Il est important de mesurer l'impact de chacun. Des tests portant sur diverses configurations des méthodes sont lancés afin de valider l'intérêt des différents éléments.

De plus, une comparaison entre les algorithmes basée uniquement sur la valeur moyenne de la qualité des solutions et/ou des temps d'exécution n'est pas toujours valable. Des performances exceptionnelles par exemple (très bonnes ou très mauvaises) sur un petit nombre d'instances influent sur la performance globale (moyenne) de l'algorithme observé. C'est pourquoi une évaluation statistique est nécessaire afin de sélectionner les meilleurs paramétrages des métaheuristiques, et c'est ce qui est réalisé sur les derniers travaux publiés.

Les hypothèses sur la distribution des résultats expérimentaux étant en question, le test non-paramétrique de Friedman est sélectionné, suivi par le test post-hoc de Bonferroni-Dunn. Ces tests sont détaillés dans les manuels de statistiques de Sheskin (2000), de Kvam et Vidakovic (2007) ou de Sprent et Smeeton (2001).

Le choix s'est limité à deux valeurs pour chacun des p principaux paramètres aboutissant à 2^p combinaisons possibles. L'hypothèse nulle considérée est que toutes les versions ont des performances égales. Le test de Friedman, exécuté à l'aide du logiciel SPSS permet d'obtenir la valeur-p permettant ou non de rejeter l'hypothèse nulle au niveau de risque α. Si l'hypothèse est rejetée, le test de Bonferroni-Dunn permet de comparer les candidats (combinaisons) avec l'ensemble des autres paramétrages et de vérifier que la combinaison choisie apporte bien une amélioration significative des résultats au niveau de probabilité choisi.

3.3.5.2 Exemples de résultats

Tous les résultats ne vont pas être présentés. Seul un résumé de quelques performances obtenues pour un sous-ensemble des problèmes étudiés est sélectionné. Il s'agit de ceux pour lesquels des améliorations significatives par rapport à la littérature permettent d'attester de la qualité des méthodes développées. Pour cela, une synthèse est proposée dans des tableaux qui ont la forme suivante. La première colonne représente les types d'instances. *Gap*, *Tps* et *NBKS* indiquent respectivement les écarts aux meilleures solutions connues, un temps de calcul en secondes sur l'ordinateur indiqué en bas de tableau et le pourcentage de meilleures solutions atteintes.

Problème de localisation-routage

Pour le problème de localisation-routage, les principales contributions sont :

- l'application d'un algorithme Split avec gestion de ressources au sein d'un algorithme hybride, noté GRASPxELS (Duhamel *et al.*, 2010a);
- une matheuristique combinant de manière coopérative une résolution par relaxation Lagrangienne et optimisation des sous-gradiants et un algorithme tabou, noté LRGTS (Prins *et al.*, 2007);
- un algorithme de branchement et coupes permettant de valider l'optimalité de certaines solutions ou proposant des bornes inférieures, noté B&C (Belenguer *et al.*, 2011).

Toutes ces méthodes sont testées sur 3 jeux d'instances dites de Barreto, de Tuzun, et celles spécifiquement développées, dites de Prodhon. Un autre jeu d'instances réalistes basées sur un graphe non complet (asymétrique) est également proposé, mais pour le moment, aucune comparaison n'est disponible avec des méthodes d'autres auteurs. Toutes ces instances sont disponibles sur : http://prodhonc.free.fr

Des comparaisons avec des résultats parus depuis la publication des méthodes proposées et qui rivalisent avec les solutions obtenues sont aussi ajoutées : VNS (Pirkwieser et Raidl, 2010) et SALRP (Yu *et al.*, 2010). D'autres travaux ont été réalisés sur le sujet, mais soit les auteurs n'utilisent pas nos instances (en particulier pour les méthodes dédiées au cas sans capacité), soit les résultats ne sont pas réellement compétitifs, soit encore des contraintes sont ajoutées comme c'est le cas pour le LRP avec pick-up and delivery où les instances Prodhon sont parfois utilisées mais dans une version modifiée.

Le Tableau 3.1 résume les résultats. Il apparaît que la plupart des instances ne sont toujours pas résolues à l'optimum par les méthodes proposées puisque le nombre de BKS (meilleures solutions connues, obtenues par divers paramétrages des algorithmes) non atteintes et l'écart aux bornes inférieures (obtenues par le B&C), sont encore relativement élevés. LRGTS, qui présentait le plus petit écart avec les BKS jusqu'en 2010 est aujourd'hui détrôné mais au coût d'un temps de calcul relativement élevé (surtout si l'on tient compte des puissances de calcul utilisées). A temps de calcul similaires, les méthodes proposées restent dominantes: **en quelques secondes, seul VNS reste compétitif avec LRGTS, avec un léger avantage pour ce dernier - en quelques minutes, GRASPxELS semble meilleur que SALRP**.

Instances		B&C 2011	LRGTS 2007	GRASPxELS 2010	VNS 2010	SALRP 2010
	Gap	-0.53%	1.66%	0.08%	/	0.29%
Barreto	Tps	1471.71s	17.58s	187.62s	/	154.05s
	NBKS	46%	15%	77%	/	77%
Note : B&C résout 10 instances sur 13 dans le temps imparti de 7200s (soit 77%)						
	Gap	/	1.45%	0.91%	/	1.10%
Tuzun	Tps	/	21.24s	606.64s	/	826.42s
	NBKS	/	0%	19%	/	14%
	Gap	-0.95%	0.71%	1.05%	0.84%	0.39%
Prodhon	Tps	3859.96s	17.48s	258.17s	6.71s	434.37s
	NBKS	20%	17%	40%	7%	37%
Note : B&C résout 12 instances sur 30 dans le temps imparti de 7200s (soit 40%)						

B&C : Intel Core 2 Quad Q6700 à 2.66 GHz avec 2 GB de RAM

LRGTS : Pentium 4 à 2.4 GHz avec 512 MB de RAM

GRASPxELS : Quad core à 2.83 GHz avec 8 GB de RAM

VNS : Intel Core 2 Quad Q9550 à 2.83 GHz avec 8 GB de RAM

SALRP : Intel Core 2 Quad à 2.6 GHz avec 2 GB de RAM

Table 3.1: Résultats pour le LRP

Problème de tournées à flotte hétérogène limitée

Concernant le problème de tournées à flotte hétérogène limitée, les trois méthodes expérimentées sont l'application d'un algorithme Split avec gestion de ressources classique, et une exploration en profondeur du graphe auxiliaire au sein d'algorithmes hybrides, ainsi qu'une version parallèle, notée respectivement GRASPxELS, GRASPx-ELS_DFS (Duhamel *et al.*, 2012a) et Parallel_ELS (Duhamel *et al.*, à paraître).

Ces méthodes sont validées sur des jeux d'instances classiques de la littérature, aussi bien pour des cas de flottes illimitées (instances VFMP) que limitées (instances HVRP). **Chacun des algorithmes développés est compétitif pour l'ensemble des variantes, alors que les autres auteurs proposent des solutions obtenues avec des méthodes dédiées** (excepté Prins (2009b) qui adapte tout de même son paramétrage). La performance observée se doit néanmoins d'être relativisée par le fait que l'écart moyen aux meilleures solutions est inférieur à 1%, voire inférieur à 0.1% selon les groupes d'instances, pour les meilleures méthodes publiées ces 5 dernières années.

De nouvelles instances sont proposées sur http://prodhonc.free.fr, basées sur un kilométrage réel entre les villes de même département français. Ces instances de petite à très grande taille ne sont pas encore suffisamment répandues dans la communauté et malheureusement aucune comparaison avec d'autres auteurs n'est actuellement disponible.

Problème de tournées avec remorques

Pour les problèmes avec remorques, les résultats rappelés sont ceux du GRASPxVND complété d'un Path Relinking, utilisant une version optimale spécifique du Split (Villegas *et al.*, 2010b).

Le Tableau 3.2 compare les résultats obtenus avec ceux de Chao (2002); Scheuerer (2006); Lin *et al.* (2009); Caramia et Guerriero (2010). La colonne n indique le nombre de clients traités. **Il apparaît clairement que le GRASPxVND proposé obtient le plus petit écart aux meilleures solutions connues (BKS)**. Les temps de calcul ne sont pas reportés dans cette table. Cependant, en mettant les différents processeurs à une même échelle, les résultats sont atteints plus rapidement (mais Caramia et Guerriero (2010) n'indiquent pas de temps de calcul).

Problem #	n	BKS	Chao Gap	Scheuerer Gap	Lin Gap	Caramia Gap	GRASPxVND Gap
1	50	564.68	0.06	0.59	0.74	0.38	0.23
2	50	611.53	8.39	1.28	0.97	1.41	0.44
3	50	618.04	7.56	1.87	0.40	2.34	0.00
4	75	798.53	7.43	1.33	2.40	0.60	0.62
5	75	839.62	13.14	2.31	2.30	0.34	0.24
6	75	930.64	16.57	2.07	1.29	0.81	3.31
7	100	830.48	0.88	0.29	0.97	0.25	0.00
8	100	872.56	3.85	1.00	1.16	0.72	0.42
9	100	912.02	9.68	4.82	1.09	7.50	0.71
10	150	1039.07	3.64	1.31	3.40	2.05	1.06
11	150	1093.37	7.02	1.29	1.42	7.07	0.69
12	150	1152.32	5.61	2.80	1.24	2.26	0.57
13	199	1287.18	6.01	0.71	4.18	0.10	1.45
14	199	1339.36	9.32	3.34	2.13	2.48	1.10
15	199	1420.72	8.69	4.79	2.41	3.48	1.18
16	120	1002.49	6.22	0.05	0.48	0.22	0.06
17	120	1026.20	7.65	1.62	0.88	1.57	1.60
18	120	1098.15	9.46	3.99	1.09	2.82	1.86
19	100	813.30	9.09	0.08	1.19	0.02	0.80
20	100	848.93	13.44	0.47	1.19	0.00	1.32
21	100	909.06	4.76	0.55	0.70	0.00	0.00
Gap moyen			7.55%	1.74%	1.51%	1.74%	0.84%
NBKS			0%	0%	0%	1%	2%

Table 3.2: Comparaison du GRASPxVND avec d'autres approches de la littérature

Problème de localisation-routage à deux niveaux

Pour les problèmes de localisation-routage à deux niveaux, aucune comparaison n'est disponible actuellement avec d'autres auteurs. Des jeux d'instances spécifiques sont proposés sur http://prodhon.free.fr, et les premières bornes inférieures développées nécessitent encore des améliorations. Cependant, il est intéressant de souligner **les excellents résultats obtenus sur le cas particulier du problème de localisation-routage**, puisque les approches développées (Nguyen *et al.*, 2012a,b) sont compétitives sans avoir de méthode dédiée. De plus, lorsque sur certaines instances le B&C trouve l'optimum, **les métaheuristiques trouvent cet optimum** également.

De plus, **ces méthodes sont facilement généralisables**. MS-ILS (Nguyen *et al.*, 2012b) a déjà été adaptée pour résoudre une version plus générale du LRP-2E dans laquelle plusieurs dépôts principaux sont disponibles. Les résultats alors obtenus sont comparés avec ceux de Boccia *et al.* (2010) et une amélioration moyenne de 3% est constatée par rapport à son approche taboue.

Partie 4

Conclusions et Perspectives

.

Après presque 10 ans dans le domaine de la recherche, d'abord avec un DEA, puis une thèse suivie d'un poste d'ATER et enfin en tant que maître de conférences à l'Université de Technologie de Troyes au sein du LOSI, ce mémoire dresse un bilan de mes activités.

Mon intérêt se porte essentiellement sur les problèmes de tournées de véhicules. Ils sont en relation avec l'intérêt grandissant de la société pour les problèmes de logistique du transport.

L'approche classique, majoritairement adoptée dans le passé pour l'optimisation des systèmes logistiques, est la décomposition des problèmes selon les niveaux de décision, en commençant par l'échelon stratégique. L'avantage d'une telle démarche est la simplification apportée. Cependant, de cette conduite résulte bien souvent une perte de l'optimalité globale car l'intégralité du problème n'est pas prise en compte. Il est donc intéressant d'adopter une vision plus large considérant plusieurs niveaux de décision.

Parmi les problèmes de transport, mes activités ont surtout porté sur ceux intégrant des composantes relatives aux choix des dépôts et aux limitations ressources, telles les capacités ou le nombre limité des véhicules.

Une première étude porte sur le **problème de localisation-routage** *LRP* combinant des capacités limitées à la fois sur les dépôts et les véhicules. Ces travaux, en collaboration avec Christian Prins et Roberto Wolfler-Calvo lors de ma thèse de doctorat, ont donné lieu à d'autres collaborations comme avec Patrick Soriano et Angel Ruiz du CIRRELT au Canada, Jose-Manuel Belenguer et Enrique Benavent de l'Université de Valencia en Espagne, ou avec Philippe Lacomme et Christophe Duhamel du LIMOS à Clermont-Ferrand.

Avec ces derniers coauteurs, nous nous intéressons également au **problème de tournées avec flotte hétérogène limitée**. La considération de cette nouvelle contrainte de ressources combinée à de la localisation amène au **problème de tournées avec remorques**. La localisation ici sert de station intermédiaire où déposer les remorques. Les tournées à élaborer sont alors à deux niveaux. Cette étude est menée à travers ma participation à la thèse en co-tutelle de Juan Guillermo Villegas, encadrée par Christian Prins et Andres Medaglia de l'Université des Andes, Colombie.

Pour continuer sur les problèmes à deux niveaux de tournées, mais en intégrant la contrainte de capacité sur les dépôts intermédiaires ou satellites, nous avons proposé l'étude de **localisation-routage à deux niveaux** dans le cadre de la thèse de Viet Phuong Nguyen co-encadrée avec Christian Prins.

En parallèle de ces recherches, je me suis intéressée à la version **multi-période du LRP** afin de m'amener au cas comportant des contraintes sur les capacités de stockage chez les clients, le **LRP avec gestion des stocks**. Ce dernier problème fait l'objet de la thèse en co-tutelle de William Guerrero que je co-encadre avec Nubia Velasco de l'Université des Andes en Colombie. Dans le même esprit, nous sommes en train d'étudier le problème de tournées avec flotte limitée et gestion des stocks avec l'équipe du LIMOS.

A travers toutes ces problèmatiques, différentes méthodes de résolutions sont développées, aussi bien exactes par les **méthodes polyédrales** qu'approchées par des **heuristiques** telles que des **méthodes hybridées** de diverses manières.

Ce qui en ressort est l'importance de l'exploration judicieuse des solutions. Pour cela, outre les caractéristiques classiques d'intensification/diversification de la recherche, l'intérêt d'**alterner entre plusieurs espaces de solutions** apparaît primordial.

Cette alternance se retrouve par exemple dans le cas avec l'espace des *tours géants* et celui des tournées par l'intermédiaire de la procédure **Split**. Diverses implémentations de cette procédure sont élaborées pour les cas d'études, en particulier avec contraintes de ressources.

De manière générale, il est intéressant d'alterner de manière intelligente entre deux espaces ou deux sous-problèmes. C'est ce qui est proposé avec la **matheuristique coopérative LRGTS** faisant interagir une résolution par relaxation Lagrangienne et optimisation des sous-gradients d'un problème de localisation et une résolution par un algorithme tabou granulaire d'un problème de tournées.

Toutes les méthodes sont validées par des séries de tests sur des instances classiques ou spécifiquement générées. Les comparaisons avec la littérature et les meilleures bornes permettent d'attester de la qualité des résultats obtenus.

L'ensemble de mes recherches a donné lieux à un chapitre de livre, 14 articles publiés dans des revues internationales et près de 40 communications dans des conférences nationales et internationales. Notons de plus qu'un chapitre de livre et un article de revue sont en révision.

De ce bilan résultent diverses **perspectives de recherche**. La première concerne les problèmatiques. Je pense continuer dans la thématique du transport, en particulier pour les problèmes intégrant diverses contraintes additionnelles qui s'approche d'avantage des problèmes réels. Avec les premiers travaux sur le LRP-2E, on peut penser à des applications comme les tournées de véhicules électriques en centre-ville qui doivent se recharger régulièrement à des bornes. Le domaine des services et la gestion des risques seront à l'avenir également très certainement des pistes à explorer plus en profondeur. Enfin, un aspect particulièrement intéressant et qui peut venir enrichir les études précédemment citées, concerne la considération de données dynamiques et/ou stochastiques.

La seconde perspective est en lien avec les méthodologies de résolution. Il me semble fondamental d'aborder les problèmes avec une vision globale mais aussi d'analyser leur structure afin de proposer une exploration judicieuse de l'espace de recherche. Les problèmes devenant de plus en plus complexes, une recherche sur l'espace complet devient prohibitoire. Cependant, un découpage, hiérarchique par exemple, aboutit de manière quasi-certaine à une sous-optimalité. Il est donc indispensable de réfléchir à une résolution passant par des problèmes simplifiés ou relaxés, mais tout en garantissant une certaine coopération pour préserver une efficacité de résolution. Il faut pas oublier la possibilité de développer des résolutions exploitant les multi-cœurs de nos ordinateurs non pas dans un simple but d'accélération de calcul, mais plutôt afin de proposer des explorations pertinentes de l'espace des solutions. Une autre piste qui me semble intéressante est la résolution multicritère. En effet, il est important de tenir compte des différents objectifs parfois antagonistes que peuvent contenir certaines problématiques réelles. J'ai déjà abordé cet aspect, mais il serait très certainement profitable d'approfondir le sujet.

Bibliographie

Ahmadi Javid, A. et Azad, N. (2010). Incorporating location, routing and inventory decisions in supply chain network design. *Transportation Research Part E: Logistics and Transportation Review*, *46*(5).

Andersson, H., Hoff, A., Christiansen, M., et Løkketangen, A. (2010). Invited review: Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, *37*, 1515–1536.

Angelelli, E. et Speranza, M. G. (2002). The periodic vehicle routing problem with intermediate facilities. *European Journal of Opearational Research*, *137*, 233–247.

Archetti, C., Bertazzi, L., Laporte, G., et Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, *41*(3).

Augerat, P., Belenguer, J.-M., Benavent, E., Corberan, A., et Naddef, D. (1999). Separating capacity constraints in the CVRP using tabu search. *European Journal of Operational Research*, *106*, 546–557.

Beasley, J. E. (1983). Route-first cluster-second methods for vehicle routing. *Omega*, *11*, 403–408.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2006a). A branch-and-cut method for the capacitated location-routing problem. In *Proceedings of IEEE/ICSSSM'06 (IEEE International Conference on Services Systems and Services Management)*. Troyes, France.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2006b). A cutting plane method for the capacitated location-routing problem. In *Odysseus 2006 (Third International Workshop on Freight Transportation and Logistics)*. Altea, Spain.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2006c). New results with integer formulations for the capacitated location-routing problem. In *EURO XXI (*21st *European Conference on Operational Research)*. Reykjavik, Island.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2006d). Some computational results with integer formulations for the location-routing problem. In *CORAL 2006 (4th Conference on Location and Routing)*. Puerto de la Cruz (Tenerife), Spain.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2011). A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, *38*(6), 931–941.

Boccia, M., Crainic, T., Sforza, A., et Sterle, C. (2010). A metaheuristic for a two echelon location-routing problem. In P. Festa (Ed.), *Lecture Notes in Computer Science*. Proceedings of SEA 2010, Springer-Verlag, vol. 6049, 288–301.

Boudia, M. et Prins, C. (2009). A memetic algorithm with dynamic population management for an integrated production-distribution problem. *European Journal of Operational Research*, *195*(3), 703–715.

Brandão, J. (2011). A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers & Operations Research*, *38*(1), 140–151.

Campbell, A. M., Clarke, L. W., et Savelsbergh, M. W. P. (2001). *Inventory routing in practice*, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics. 309–330.

Caramia, M. et Guerriero, F. (2010). A heuristic approach for the truck and trailer routing problem. *Journal of the Operational Research Society*, *61*, 1168–1180.

Çetinkaya, S. et Lee, C.-Y. (2000). Stock replenishment and shipment scheduling for vendor-managed inventory systems. *Management Science*, *46*(2), 217–232.

Chao, I.-M. (2002). A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, *29*(1), 33–51.

Chao, I.-M., Golden, B. L., et Wasil, E. (1995). An improved heuristic for the periodic vehicle routing problem. *Networks*, *26*(1), 25–44.

Choi, E. et Tcha, D.-W. (2007). A column generation approach to the heterogeneous fleet vehicle routing. *Computers & Operations Research*, *34*, 2080–2095.

Christofides, N. et Beasley, J. E. (1984). The period routing problem. *Networks*, *14*, 237–256.

Chu, F., Labadi, N., et Prins, C. (2006). A scatter search for the periodic capacitated arc routing problem. *European Journal of Operational Research*, *169*, 586–605.

Claassen, G. et Hendriks, T. (2007). An application of special ordered sets to a periodic milk collection problem. *European Journal of Operational Research*, *180*, 754–769.

Clarke, G. et Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, *12*, 568–581.

Cordeau, J.-F., Gendreau, M., et Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, *30*, 105–119.

Cordone, R., Ferrandi, F., Sciuto, D., et Wolfler-Calvo, R. (2001). An efficient heuristic approach to solve the unate covering problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *20*, 1377–1388.

Crainic, T., Mancini, S., Perboli, G., et Tadei, R. (2008). Heuristics for the two-echelon capacitated vehicle routing problem. Tech. Rep. 46, CIRRELT.

Crevier, B., Cordeau, J.-F., et Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, *176*, 756–773.

Desrochers, M. (1988). An algorithm for the shortest path problem with resource constraints. Research report G-88-27, GERAD, Montreal, Canada.

Dolgui, A., Ould Louly, M.-A., et Prodhon, C. (2005). *Selected Plenaries, Milestones and Surveys (16th IFAC World Congress, 3-8/07/2005, Prague)*, Elsevier, chap. A survey on supply planning under uncertainties in MRP environments. 228–239.

Dolgui, A. et Prodhon, C. (2007). Supply planning under uncertainties in MRP environments: a state of the art. *Annual Review on Control*, *31*(2), 269–279.

Duhamel, C., Gouinaud, C., Lacomme, P., et Prodhon, C. (à paraître). *Hybrid Metaheuristics*, Springer, chap. A multi-thread GRASPxELS for the heterogeneous capacitated vehicle routing problem.

Duhamel, C., Lacomme, P., Prins, C., et Prodhon, C. (2008). A memetic approach for the capacitated location routing problem. In *EU/MEeting 2008 (International Workshop on Metaheuristics for Logistics and Vehicle Routing)*. Troyes, France.

Duhamel, C., Lacomme, P., Prins, C., et Prodhon, C. (2009). A GRASP×ELS approach for real-life location routing problems. In *CIE39 (International Conference on Computers & Industrial Engineering)*. Troyes, France.

Duhamel, C., Lacomme, P., Prins, C., et Prodhon, C. (2010a). A GRASPxELS approach for the capacitated location-routing problem. *Computers & Operations Research*, *37*(11), 1912–1923.

Duhamel, C., Lacomme, P., et Prodhon, C. (2010b). Définition d'un split en profondeur pour les problèmes de tournées. In *Congrès ROADEF 2010*. Toulouse, France.

Duhamel, C., Lacomme, P., et Prodhon, C. (2011). Efficient frameworks for greedy split and new depth first search split procedures for routing problems. *Computers & Operations Research*, *38*(4), 723–739.

Duhamel, C., Lacomme, P., et Prodhon, C. (2012a). A hybrid evolutionary local search with depth first search split procedures for the heterogeneous vehicle routing problem. *Engineering Applications of Artificial Intelligence 25(2)*, *25*(2), 345–358.

Duhamel, C., Lacomme, P., et Prodhon, C. (2012b). Problème de tournées de véhicules à flotte hétérogène limitée et gestion des stocks intégrée. In *Congrès ROADEF 2012*. Angers, France.

Duhamel, C., Lacomme, P., Prodhon, C., et Prins, C. (2010c). Parallel cooperative GRASP for the heterogeneous vehicle routing problem. In *EURO XXIV (24st European Conference On Operational Research)*. Lisbon, Portugal.

Ehrgott, M. et Gandibleux, X. (2000). A survey and annoted bibliography of multiobjective combinatorial optimization. *OR Spektrum*, *22*, 425–460.

El Hayek, J., Moukrim, A., et Negre, S. (2008). New resolution algorithm and pretreatments for the two-dimensional bin-packing problem. *Computers & Operations Research*, *35*(10), 3184–3201.

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.

Gao, L.-L. et Robinson Jr., E. P. (1992). A dual-based optimization procedure for the two-echelon uncapacitated facility location problem. *Naval Research Logistics*, *39*, 191–212.

Gendron, B. et Semet, F. (2009). Formulations and relaxations of a multi-echelon capacitated location-distribution problem. *Computers & Operations Research*, *36*, 1335–1355.

Gerdessen, J. C. (1996). Vehicle routing problem with trailers. *European Journal of Operational Research*, *93*(1), 135–147.

Golden, B. L., Assad, A., Levy, L., et Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, *11*, 49–66.

Gonzalez-Feliu, J. (2008). *Models and methods for the city logistics – The two-echelon capacitated vehicle routing problem*. Ph.D. thesis, Politecnico di Torino, Italy.

Gonzalez-Feliu, J. (2010). The multi-echelon location-routing problem: Concepts and methods for tactical and operational planning. Tech. rep., Laboratoire d'Economie des Transports, Lyon, France.

Guerrero, W. J., Prodhon, C., Velasco, N., et Amaya, C. A. (2011a). A matheuristic for the inventory location routing problem with deterministic demand. In *MIC 2011 (Metaheuristics International Conference)*. Udine, Italie.

Guerrero, W. J., Prodhon, C., Velasco, N., et Amaya, C.-A. (2012a). Hybrid heuristic for the inventory location-routing problem. In *Odysseus'12 (5^{th} International Workshop on Freight Transportation and Logistics)*. Mykonos, Greece.

Guerrero, W. J., Prodhon, C., Velasco, N., et Amaya, C.-A. (2012b). Integration d'une gestion de stocks lors de la résolution coopérative du problème de localisation-routage. In *Congrès ROADEF 2012*. Angers, France.

Guerrero, W. J., Velasco, N., Amaya, C.-A., et Prodhon, C. (2011b). The inventory location routing problem with deterministic demand. In *POM 2011 (22^{nd} Annual Conference of the Production and Operations Management Society)*. Reno (Nevada), USA.

Hemmelmayr, V., Doerner, K., et Hartl, R. F. (2009). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, *195*(3), 791–802.

Hoff, A. et Løkketangen, A. (2007). A tabu search approach for milk collection in western norway using trucks and trailers. In *TRISTAN VI (The sixth triennial symposium on transportation analysis)*. Phuket, Thailand.

Jacobsen, S. K. et Madsen, O. B. G. (1980). A comparative study of heuristics for a two-level routing-location problem. *European Journal of Operational Research*, *6*, 378–387.

Johnson, D. S. et Papadimitriou, C. H. (1985). Computational complexity. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, et D. B. Shmoys (Eds.), *The traveling Salesman Problem*. Chichester: John Wiley & Sons, 37–86.

Jordan, W. C. et Burns, L. D. (1984). Truck backhauling on two terminal networks. *Transportation Research*, *18B*, 487–503.

Kaufman, L., Ecde, M., et Hansen, P. (1977). A plant and warehouse location problem. *Operational Research Quarterly*, *28*, 547–554.

Kébé, S., Sbihi, N., et Penz, B. (2012). A lagrangean heuristic for a two-echelon storage capacitated lot-sizing problem. *Journal of Intelligent Manufacturing*, *A paraître*.

Kvam, P. et Vidakovic, B. (2007). *Nonparametric Statistics with Applications to Science and Engineering*. New Jersey: Wiley.

Lacomme, P., Prins, C., et Ramdane-Chérif, W. (2004). Competitive memetic algorithms for arc routing problems. *Annals of Operations Research*, *131*, 159–185.

Lacomme, P., Prins, C., et Ramdane-Chérif, W. (2005). Evolutionary algorithms for periodic arc routing problems. *European Journal of Operational Research*, *165*(2), 535–553.

Lacomme, P. et Prodhon, C. (2010). Parallel cooperative ELS for the heterogeneous vehicle routing problem. In *EU/MEeting 2010 (10^{th} Anniversary of the Metaheuristics Community)*. Lorient, France.

Laporte, G., Norbert, Y., et Arpin, D. (1984). Optimal solutions to capacitated multi-depot vehicle routing problems. *Congressus Numerantium*, *44*, 283–292.

Laporte, G., Norbert, Y., et Arpin, D. (1986). An exact algorithm for solving a capacitated location-routing problem. *Annals of Operations Research*, *6*, 293–310.

Laporte, G., Norbert, Y., et Taillefer, S. (1988). Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Science*, *22*, 161–167.

Levy, L. et Bodin, L. (2000). *Arc Routing: Theory, Solutions and Applications*, Dordrecht: Kluwer, chap. Scheduling of local delivery carriers for the united states postal service. 419–442.

Li, X., Tian, P., et Aneja, Y. P. (2010). An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Transportation Research Part E*, *46*(6), 1111–1127.

Lin, S. et Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, *21*, 498–516.

Lin, S.-W., Yu, V. F., et Chou, S.-Y. (2009). Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, *36*(5), 1683–1692.

Liu, S. et Lee, S. (2003). A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. *The International Journal of Advanced Manufacturing Technology*, *22*, 941–950.

Liu, S. et Lin, C. (2005). A heuristic method for the combined location routing and inventory problem. *The International Journal of Advanced Manufacturing Technology*, *26*, 372–381.

Lysgaard, J., Letchford, A. N., et Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, *100*, 423–445.

Min, H., Jayaraman, V., et Srivastava, R. (1998). Combined location-routing problems: A synthesis and future research directions. *European Journal of Operational Research*, *108*, 1–15.

Nagy, G. et Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research*, *177*, 649–672.

Nguyen, V.-P., Prins, C., et Prodhon, C. (2010a). Approches heuristiques pour le problème de localisation - routage à deux niveaux. In *Congrès ROADEF 2010*. Toulouse, France.

Nguyen, V.-P., Prins, C., et Prodhon, C. (2010b). GRASP with learning process for a two-echelon location routing problem. In *MOSIM'10 (8^{e} Conférence Francophone de Modélisation et Simulation)*. Hammamet, Tunisia, 10p.

Nguyen, V.-P., Prins, C., et Prodhon, C. (2010c). A hybrid GRASP x path relinking for the two-echelon location routing problem. In *EURO XXIV (24st European Conference On Operational Research)*. Lisbon, Portugal.

Nguyen, V.-P., Prins, C., et Prodhon, C. (2010d). A multi-start evolutionary local search for the two-echelon location routing problem. In M. J. Blesa, C. Blum, G. Raidl, A. Roli, et M. Sampels (Eds.), *Lecture Notes in Computer Science*. Proceedings of Hybrid Metaheuristics 2010, Vienne, Austria, Octobre, 2010, Springer, vol. 6373, 88–102.

Nguyen, V.-P., Prins, C., et Prodhon, C. (2011a). A cutting plane method for the two-echelon location routing problem. In *Optimization 2011*. Lisbon, Portugal.

Nguyen, V.-P., Prins, C., et Prodhon, C. (2011b). A hybrid metaheuristic for the two-echelon location routing problem. In *IESM 2011 (International Conference on Industrial Engineering and Systems Management)*. Metz, France.

Nguyen, V.-P., Prins, C., et Prodhon, C. (2012a). A multi-start iterative local search with tabu list and path relinking for the two-echelon location routing problem. *Engineering Applications of Artificial Intelligence*, *25*(1).

Nguyen, V.-P., Prins, C., et Prodhon, C. (2012b). Solving the two-echelon location routing problem by a hybrid GRASP x path relinking complemented by a learning process. *European Journal of Operational Research*, *216*, 113–126.

Nguyen, V.-P., Prins, C., Prodhon, C., et Villegas, J. G. (2012c). A branch-and-cut algorithm for the two-echelon location-routing problem. In *Odysseus'12 (5th International Workshop on Freight Transportation and Logistics)*. Mykonos, Greece.

OMC (2012). http://www.wto.org/french/tratop_f/envir_f/climate_impact_f.htm.

Osman, I. H. et Salhi, S. (1996). *Modern Heuristic Search Methods*, Chichester: Wiley, chap. Local search strategies for the mix fleet routing problem. 131–153.

Padberg, M. et Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, *33*, 60–100.

PIPAME (2009). Logistique et distribution urbaine.

Pirkwieser, S. et Raidl, G. R. (2010). Variable neighborhood search coupled with ilp-based very large neighborhood searches for the (periodic) location-routing problem. In M. J. Blesa, C. Blum, G. Raidl, A. Roli, et M. Sampels (Eds.), *Hybrid Metaheuristics, Lecture Notes in Computer Science*. Proceedings of Hybrid Metaheuristics 2010, Vienne, Austria, Octobre, 2010, Springer-Verlag Heidelberg, vol. 6373, 174–189.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, *31*, 1985–2002.

Prins, C. (2009a). *Bio-inspired Algorithms for the Vehicle Routing Problem*, Springer, vol. 161 of *Studies in Computational Intelligence*, chap. A GRASP x Evolutionary Local Search Hybrid for the Vehicle Routing Problem. 35–53.

Prins, C. (2009b). Two memetic algorithms for heterogeonous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, *22*, 916–928.

Prins, C. et Prodhon, C. (2008). A memetic algorithm with population management (*MA* | *PM*) for the periodic location-routing problem. In M. J. Blesa, C. Blum, C. Cotta, A. J. Fernández, J. E. Gallardo, A. Roli, et M. Sampels (Eds.), *Lecture Notes in Computer Science*. Proceedings of Hybrid Metaheuristics 2008, Málaga, Espagne, Septembre, 2008, Springer, vol. 5296, 43–57.

Prins, C., Prodhon, C., Ruiz, A., Soriano, P., et Wolfler-Calvo, R. (2005a). A cooperative lagrangean relaxation-granular tabu search heuristic for the capacitated location routing problem. In *MIC'05 (Metaheuristics International Conference)*. Vienna, Austria.

Prins, C., Prodhon, C., Ruiz, A., Soriano, P., et Wolfler-Calvo, R. (2005b). A new metaheuristic combining a lagrangean relaxation with a tabu search approach for the location-routing problem. In *AIRO'05 (Annual Conference of the Italian Operational Research Society)*. Camerino, Italy.

Prins, C., Prodhon, C., Soriano, P., Ruiz, A., et Wolfler-Calvo, R. (2007). Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. *Transportation Science*, *41*(4), 470–483.

Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2004). Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité. In A. Dolgui et S. Dauzère-Pérès (Eds.), *MOSIM'04 (Conférence Francophone de Modélisation et Simulation)*. Ecole des Mines de Nantes: Lavoisier, vol. 2, 1115–1122.

Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2005c). A reactive GRASP and path relinking algorithm for the capacitated location routing problem. In *IESM'05 (International Conference on Industrial Engineering and Systems Management)*. Marrakech, Morocco: I4E2.

Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2006a). Comparaisons de méthodes de résolution pour le problème de localisation-routage sous contraintes de capacité. In *Congrès ROADEF 2006*. Lille, France.

Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2006b). A memetic algorithm with population management ($MA \mid PM$) for the capacitated location-routing problem. In J.Gottlieb et G. R. Raidl (Eds.), *Lecture Notes in Computer Science*. Proceedings of EvoCOP2006 (Evolutionary Computation in Combinatorial Optimization: 6th European Conference, Budapest, Hungary, April 10-12, 2006), Springer, vol. 3906, 183–194.

Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2006c). Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *4OR - A Quarterly Journal of Operations Research*, *4*, 221–238.

Prins, C., Prodhon, C., et Wolfler-Calvo, R. (2006d). Two-phase method and lagrangian relaxation to solve the bi-objective set covering problem. *Annals of Operations Research*, *147*, 23–41.

Prodhon, C. (2006). Approche coopérative pour le problème de localisation-routage. In *Journées STP du GdR MACS*. Valenciennes, France.

Prodhon, C. (2007). Solving the capacitated location-routing problem. *4OR - A Quarterly Journal of Operations Research*, *5*, 339–342.

Prodhon, C. (2008a). Etude du problème de localisation-routage périodique. In *Congrès ROADEF 2008*. Clermont-Ferrand, France.

Prodhon, C. (2008b). A metaheuristic for the periodic location-routing problem. In *GOR'07 (Operations Research 2007)*. Saarbrücken, Allemagne.

Prodhon, C. (2009a). An ELS x path relinking hybrid for the periodic location-routing problem. In *Journée du GT Transport et Logistique*. Le Havre, France.

Prodhon, C. (2009b). An ELS x path relinking hybrid for the periodic location-routing problem. In M. J. Blesa, C. Blum, L. Di Gaspero, A. Roli, M. Sampels, et A. Schaerf (Eds.), *Lecture Notes in Computer Science*. Proceedings of Hybrid Metaheuristics 2008, Udine, Italie, Octobre, 2009), Springer, vol. 5818, 15–29.

Prodhon, C. (2009c). An evolutionary algorithm for the periodic location-routing problem. In *Odysseus'09 (4th International Workshop on Freight Transportation and Logistics)*. ζesme, Turquie.

Prodhon, C. (2010). Solution methods for the periodic location-routing problem. In *EURO XXIV (24st European Conference On Operational Research)*. Lisbon, Portugal.

Prodhon, C. (2011). A hybrid evolutionary algorithm for the periodic location-routing problem. *European Journal of Operational Research*, *210*(2).

Prodhon, C., Macias, D., Yalaoui, F., Vial, A., et Amodeo, L. (2009a). Bi-objective optimization of the plasmon-assisted lithography - design of plasmonic nanostructures. In *International joint conference on computational intelligence*. Funchal, Portugal, 4p.

Prodhon, C., Macias, D., Yalaoui, F., Vial, A., et Amodeo, L. (2009b). Evolutionary optimization for plasmon-assisted lithography. In M. Giacobini, A. Brabazon, S. Cagnoni, A. Ekart, A. Esparcia-Alcázar, M. Farooq, A. Fink, P. Machado, J. McCormack, M. O'Neill, F. Neri, M. Preuss, F. Rothlauf, E. Tarantino, et S. Yang (Eds.), *Lecture Notes in Computer Science*. Applications of Evolutionary Computing (Proceedings of EvoWorkshops 2009 (EvoIASP 2009), Tübingen, Germany, April, 2009), Springer Verlag, vol. 5484, 420–425.

Ro, H. et Tcha, D. (1984). A branch and bound algorithm for the two-echelon uncapacited facility location problem with some side constraints. *European Journal of Opearational Research*, *18*, 349–358.

Salhi, S. et Rand, G. K. (1989). The effect of ignoring routes when locating depots. *European Journal of Operational Research*, *39*, 150–156.

Scheuerer, S. (2006). A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, *33*, 894–909.

Semet, F. et Taillard, E. (1993). Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research*, *41*(4).

Sheskin, J. (2000). *Handbook of parametric and non-parametric statistical procedures*. New York: Chapman & Hall/CRC.

Sörensen, K. et Sevaux, M. (2006). *MA | PM*: memetic algorithms with population management. *Computers & Operations Research*, *33*, 1214–1225.

Sprent, I. et Smeeton, N. (2001). *Applied nonparametric statistical methods*. New York: Chapman & Hall/CRC.

Taillard, E. D. (1999). A heuristic column generation method for the heterogeneous fleet vehicle routing problem. *RAIRO Operations Research*, *33*(1), 1–14.

Tan, C. et Beasley, J. E. (1984). A heuristic algorithm for the periodic vehicle routing problem. *OMEGA International Journal of Management Science*, *12*(5), 497–504.

Toth, P. et Vigo, D. (Eds.) (2001). *The vehicle routing problem*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Tragantalerngsak, S., Holt, J., et Ronnqvist, M. (2007). Lagrangian heuristics for the two-echelon, single-source, capacitated facility location problem. *European Journal of Operational Research*, *102*, 611–625.

Tuyttens, D., Teghem, J., Fortemps, P., et Nieuwenhuyze, K. V. (2000). Performance of the MOSA method for the bicriteria assignment problem. *Journal of Heuristics*, *6*, 295–310.

Villegas, J. G., Belenguer, J.-M., Benavent, E., Prins, C., et Prodhon, C. (2010a). A cutting plane approach for the single truck and trailer routing problem with satellite depots. In *EURO XXIV (24st European Conference On Operational Research)*. Lisbon, Portugal.

Villegas, J. G., Medaglia, A. L., Mendoza, J., Prins, C., Prodhon, C., et Velasco, N. (2008). A split-based framework for the vehicle routing problem. In *CLAIO 2008 (XIV Congreso Latino Ibero Americano de Investigacion de Operaciones)*. Catagena, Colombia.

Villegas, J. G., Medaglia, A. L., Prins, C., Prodhon, C., et Velasco, N. (2009a). GRASP/evolutionary local search hybrids for a truck and trailer routing problem. In *MIC 2009 (8th Metaheuristics International Conference)*. Humbourg, Germany.

Villegas, J. G., Medaglia, A. L., Prins, C., Prodhon, C., et Velasco, N. (2009b). Metaheuristics for a truck and trailer routing problem. In *CORAL 2009 (5th Conference on Routing and Location)*. Elche, Spain.

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., et Velasco, N. (2010b). GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, *23*(5).

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., et Velasco, N. (2010c). GRASP/VND with path relinking for the truck and trailer routing problem. In *TRISTAN VII (7th triennial symposium on transportation analysis)*. Tromsø, Norway.

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., et Velasco, N. (2011a). GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, *38*(9).

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., et Velasco, N. (2011b). Heuristic column generation for the truck and trailer routing problem. In *IESM'11 (International Conference on Industrial Engineering and Systems Management)*. Metz, France.

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., et Velasco, N. (2011c). Heuristic column generation for the truck and trailer routing problem. In *ROUTE 2011*. Sitges, Spain.

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., et Velasco, N. (en révision). A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*.

Visee, M., Teghem, J., Pirlot, M., et Ulungu, E. (1998). Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. *Journal of Global Optimization*, *12*, 139–155.

Wang, C., Ma, Z., et Li, H. (2008). Stochastic dynamic location-routing-inventory problem in closed-loop logistics system for reusing end-of-use products. In *ICICTA 2008 (International Conference on Intelligent Computation Technology and Automation)*. Washington, DC, USA, vol. 2, 691–695.

Wu, T. H., Low, C., et Bai, J. W. (2002). Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, *29*, 1393–1415.

Xuefeng, W. (2010). An integrated multi-depot location-inventory-routing problem for logistics distribution system planning of a chain enterprise. In *ICLSIM (International Conference on Logistics Systems and Intelligent Management)*. vol. 3, 1427–1431.

Yu, V. F., Lin, S.-W., Lee, W., et Ting, C.-J. (2010). A simulated annealing heuristic for the capacitated location routing problem. *Computers & Industrial Engineering*, *58*, 288–299.

Zhang, B., Ma, Z., et Jiang, S. (2008). Location-routing-inventory problem with stochastic demand in logistics distribution systems. In *WiCOM 2008 (4^{th} International Conference on Wireless Communications, Networking and Mobile Computing)*. 1–4.

Zhao, Q., Chen, S., Leung, S. C. H., et Lai, K. K. (2010). Integration of inventory and transportation decisions in a logistics system. *Transportation Research Part E: Logistics and Transportation Review*, *46*(6), 913–925.

Partie 5

Annexe

Sélection de quatre articles :

- C. Prins, C. Prodhon, P. Soriano, A. Ruiz, R. Wolfler Calvo, Solving the Capacitated Location-Routing Problem by a Cooperative Lagrangean Relaxation-Granular Tabu Search Heuristic, *Transportation Science*, 41(4), pp. 470-483, 2007.
- C. Duhamel, P. Lacomme, C. Prodhon. Efficient frameworks for greedy split and new depth first search split procedures for routing problems. *Computers & Operations Research*, 38(4), pp. 723-739, 2011.
- J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. Wolfler-Calvo. A Branch-and-Cut method for the Capacitated Location-Routing Problem. *Computers & Operations Research*, 38(6), pp. 931-941, 2011.
- J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco. GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 38(9), pp. 1319-1334, 2011.

.

TRANSPORTATION SCIENCE
Vol. 41, No. 4, November 2007, pp. 470–483
ISSN 0041-1655 | EISSN 1526-5447 | 07 | 4104 | 0470

informs®

DOI 10.1287/trsc.1060.0187

Solving the Capacitated Location-Routing Problem by a Cooperative Lagrangean Relaxation-Granular Tabu Search Heuristic

Christian Prins, Caroline Prodhon
Institute Charles Delaunay, FRE CNRS 2848, Université de Technologie de Troyes, BP 2060,
10010 Troyes Cedex, France {christian.prins@utt.fr, caroline.prodhon@utt.fr}

Angel Ruiz
Département opérations et systèmes de dècision, Faculté des sciences de l'administration,
Université Laval, Québec, Canada G1K 7P4, angel.ruiz@fas.ulaval.ca

Patrick Soriano
Méthodes quantitatives de gestion, École des hautes études commercioles (HEC-Montréal),
3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7, Centre interuniversitaire
de recherche sur les réseaux d'entreprise la logistique et le transport (CIRRELT), patrick@crt.umontreal.ca

Roberto Wolfler Calvo
Institute Charles Delaunay, FRE CNRS 2848, Université de Technologie de Troyes, BP 2060,
10010 Troyes Cedex, France, roberto.wolfler@utt.fr

Most of the time in a distribution system, depot location and vehicle routing are interdependent, and recent studies have shown that the overall system cost may be excessive if routing decisions are ignored when locating depots. The location-routing problem (LRP) overcomes this drawback by simultaneously tackling location and routing decisions. This paper presents a cooperative metaheuristic to solve the LRP with capacitated routes and depots. The principle is to alternate between a depot location phase and a routing phase, exchanging information on the most promising edges. In the first phase, the routes and their customers are aggregated into supercustomers, leading to a facility-location problem, which is then solved by a Lagrangean relaxation of the assignment constraints. In the second phase, the routes from the resulting multidepot vehicle-routing problem (VRP) are improved using a granular tabu search (GTS) heuristic. At the end of each global iteration, information about the edges most often used is recorded to be used in the following phases. The method is evaluated on three sets of randomly generated instances and compared with other heuristics and a lower bound. Solutions are obtained in a reasonable amount of time for such a strategic problem and show that this metaheuristic outperforms other methods on various kinds of instances.

Key words: location-routing problem; Lagrangean relaxation; granular tabu search
History: Received: November 2005; revision received: June 2006; accepted: November 2006.

Introduction

Logistics costs often represent a large portion of company expenses. To reduce them, depot location and vehicle routing are crucial choices. Most of the time, these two levels of decision are tackled separately, but, unfortunately, it has been shown that this strategy often leads to suboptimal solutions (Salhi and Rand 1989). The location-routing problem (LRP) integrates these two decision levels. As shown in Min, Jayaraman, and Srivastava (1998), most of the early published papers consider either capacitated routes or capacitated depots (Laporte, Norbert, and Taillefer 1988; Chien 1993; Srivastava 1993), but not both. In general, the LRP is formulated as a deterministic node-routing problem (i.e., customers are located on nodes of the network). However, a few authors have studied stochastic cases (Laporte, Louveaux, and Mercure 1989; Chan, Carter, and Burnes 2001) and, more recently, arc routing versions (Ghiani and Laporte 2001; Labadi 2003).

Albareda-Sambola, Díaz, and Fernández (2005) propose a two-phase tabu search (TS) heuristic for the LRP with one single route per capacitated open depot. The algorithm consists of an intensification phase that optimizes the routes and a diversification phase that modifies the set of open depots. The method is tested on small instances only (at most 30 customers). Tuzun and Burke (1999) develop another two-phase TS, but for the LRP with capacitated routes and uncapacitated depots, that is, a depot may have as many routes as desired. The two phases are also dedicated to routing and location. The principle is to increase progressively the number of open depots until this deteriorates the total cost. These authors report results for up to 200 customers.

Hereafter, the case with capacities on both depots and routes will be referred to as the *general LRP*. Wu, Low, and Bai (2002) study the general LRP with homogeneous or limited heterogeneous fleets. They divide the original problem into two subproblems: a location-allocation problem, where each route is viewed as one single node whose demand is equal to the amount delivered by this route, and a vehicle-routing problem (VRP). Each subproblem is solved in a sequential and iterative manner by a simulated annealing algorithm with a tabu list to avoid cycling. They test their method on small- and medium-size instances (from 12 to 85 customers, but mainly instances with 50 nodes). Prins, Prodhon, and Wolfler Calvo (2005, 2006) also study the general LRP, but with a homogeneous and unlimited fleet. They propose a greedy randomized adaptive search procedure (GRASP) with a memory of the depots used during a diversification phase. This information guides the search of the most promising depots during an intensification phase. The method then ends with a postoptimization performed by a path-relinking algorithm. Results are given for up to 200 customers.

A heuristic for a three-level general LRP including factories, depots, and customers and a maximum duration per route are developed by Bruns and Klose (1996). They propose an iterative "location first-route second" heuristic in which cost coefficients for serving customers from a depot are used in the location phase and reestimated after each routing phase. Hence, the location subproblem is solved independently using a Lagrangean heuristic based on the relaxation of the capacity constraints. From the obtained assignment of customers, a tour construction procedure is executed and followed by an iterative tour improvement heuristic. The overall procedure iterates until the estimate of the cost coefficients is stabilized or a maximum number of iterations is reached. The heuristic is evaluated on instances having up to 10 plants, 50 potential depots, and 100 customers. Barreto (2004b) develops a family of three-phase heuristics based on clustering techniques. Clusters of customers fitting vehicle capacity are formed in the first phase. A traveling salesman problem (TSP) is solved for each cluster in the second phase. Finally, in the third phase, the depots to be opened are determined by solving a facility-location problem, in which the TSP cycles are aggregated into supernodes. Barreto also proposes a lower bound, which is reached by the best heuristics on some small-scale instances (up to 36 customers and 5 depots).

The present paper deals with the general LRP with fixed costs whenever a depot or a route is opened. The objective is to determine the set of depots to open and the routes originating from each of them to minimize a total cost comprised of the setup costs of depots and routes and the total variable cost of the routes. Given that the problem is NP-hard, the use of an exact approach is inconceivable for the targeted large-scale instances. Therefore, the proposed method is an approximate algorithm. It alternates between a facility-location phase, solved by a Lagrangean relaxation approach, and a routing phase, handled by a granular tabu search (GTS). This basis scheme works in synergy with a natural exchange of information. Nevertheless, some diversification is needed to escape from local optima. New solutions are built in a guided procedure using more and more information concerning the most promising edges while the algorithm moves on. This further cooperation can be seen as a branching zero adding the most promising edges in the solution. Moreover, to the best of our knowledge, this is the first time that a relaxation on the assignment constraints and a GTS are applied to LRP within a cooperative approach.

The paper is organized as follows. Section 1 defines the problem, introduces some required notation, and proposes an integer linear programming model. The depot-location phase solved by a Lagrangean relaxation is then presented in §2, followed by a description of the GTS in §3. Section 4 explains the cooperative step linking these two basic components. To evaluate the performance of the method, computational experiments are presented in §5. Some concluding remarks close the paper.

1. Problem Definition and Mathematical Model

The LRP studied in this paper is defined on a complete, weighted, and undirected network $G = (V, E, C)$. V is a set of nodes comprised of a subset I of m possible depot locations and a subset $J = V \backslash I$ of n customers. The traveling cost between any two nodes i and j is given by c_{ij}. A capacity W_i and an opening cost O_i are associated with each depot site $i \in I$. Each customer $j \in J$ has a demand d_j. A set K of identical vehicles of capacity Q is available. When used, each vehicle incurs a fixed cost F and performs a single route. The total number of vehicles used (or routes performed) is a decision variable.

The following constraints must hold:

- each demand d_j must be served by one single vehicle;
- each route must begin and end at the same depot, and its total load must not exceed vehicle capacity;
- the total load of the routes assigned to a depot must fit the capacity of that depot.

The total cost of a route includes the fixed cost F and the costs of traversed edges (variable costs). The objective is to find which depots should be opened and which routes should be constructed to minimize the total cost (fixed costs of depots plus total variable cost of the routes). Euclidean instances like those considered in §5 correspond to a complete directed graph G in which c_{ij} is the Euclidean distance between nodes i and j.

The problem can be modelled as a zero-one linear program. The following Boolean variables are used: $y_i = 1$ iff depot i is opened, $f_{ij} = 1$ iff customer j is assigned to depot i, and $x_{jlk} = 1$ iff edge (j, l) is traversed from j to l in the route performed by vehicle $k \in K$.

$$\min z = \sum_{i\in I} O_i y_i + \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} c_{ij}x_{ijk} + \sum_{k\in K}\sum_{i\in I}\sum_{j\in J} F x_{ijk}, \quad (1)$$

subject to

$$\sum_{k\in K}\sum_{i\in V} x_{ijk} = 1 \quad \forall j \in J \quad (2)$$

$$\sum_{j\in J}\sum_{i\in V} d_j x_{ijk} \le Q \quad \forall k \in K \quad (3)$$

$$\sum_{j\in V} x_{ijk} - \sum_{j\in V} x_{jik} = 0 \quad \forall k \in K, \forall i \in V \quad (4)$$

$$\sum_{i\in I}\sum_{j\in J} x_{ijk} \le 1 \quad \forall k \in K \quad (5)$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijk} \le |S| - 1 \quad \forall S \subseteq J, \forall k \in K \quad (6)$$

$$\sum_{u\in J} x_{iuk} + \sum_{u\in V\setminus\{j\}} x_{ujk} \le 1 + f_{ij} \quad \forall i \in I, \forall j \in J, \forall k \in K \quad (7)$$

$$\sum_{j\in J} d_j f_{ij} \le W_i y_i \quad \forall i \in I \quad (8)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in V, \forall j \in V, \forall k \in K \quad (9)$$

$$y_i \in \{0, 1\} \quad \forall i \in I \quad (10)$$

$$f_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in V \quad (11)$$

The objective function (1) sums all the costs described before. Constraints (2) guarantee that every customer belongs to one and only one route and that each customer has only one predecessor in the tour. Capacity constraints are satisfied through inequalities (3) and (8). Constraints (4) and (5) ensure the continuity of each route and a return to the depot of origin. Constraints (6) are subtour elimination constraints. Constraints (7) specify that a customer can be assigned to a depot only if a route linking them is opened. Finally, constraints (9), (10), and (11) state the Boolean nature of the decision variables.

The LRP is obviously NP-hard because it reduces to the well-known VRP when $m = 1$. It is much more combinatorial than the VRP because, in addition to the partition of customers into routes and the sequencing of each route, it involves the selection of open depots and the assignment of routes to these depots. Therefore, only very small instances can be solved exactly by commercial linear programming (LP) solvers, and basic relaxations such as the linear relaxation provide very weak lower bounds. A first nontrivial bound for the LRP with capacitated routes and depots has been proposed only recently (Barreto 2004b). Its quality has been proven on small instances, but the author reports no gap beyond 50 customers because it becomes too time consuming.

Due to the size of the instances targeted in this paper (up to 20 depot sites and 200 customers), a new metaheuristic is proposed. Searching a neighborhood that simultaneously allows opening or closing depots, as well as moving customers, may be very difficult and time consuming to handle. This is why we propose a solution procedure in which each global iteration performs one location phase (diversification) and one routing phase (intensification). Therefore, the method alternates between these two phases, described, respectively, in §§2 and 3, and an exchange of information between these two components is applied to create a powerful cooperative approach, which is detailed in §4.

2. Location Phase Based on Lagrangean Relaxation

The location phase begins from a feasible solution, obtained either by a constructive heuristic (at the first global iteration of the proposed algorithm) or resulting from the previous routing phase (in the subsequent iterations). From this solution, the customers from each route k and their demand are aggregated into one supercustomer k of demand D_k (Figure 1 gives an example of aggregation of customers). Its distance C_{ik} to a depot i is computed as the minimum insertion cost of this depot between two consecutive customers j and l of the original route as follows:

$$C_{ik} = c_{ij} + c_{il} - c_{jl}.$$

2.1. Facility-Location Problem

The aggregation of routes into supercustomers results in a single-source capacitated facility-location problem, where each supercustomer can only be served from a single capacitated depot. Given a set of m potential locations for source facilities (depots) and a set of supercustomers, the objective is to minimize the total cost to open depots and assign the supercustomers to them while respecting depot capacity constraints. The problem can be modelled as follows, where the binary decision variables X_{ik} equal one iff

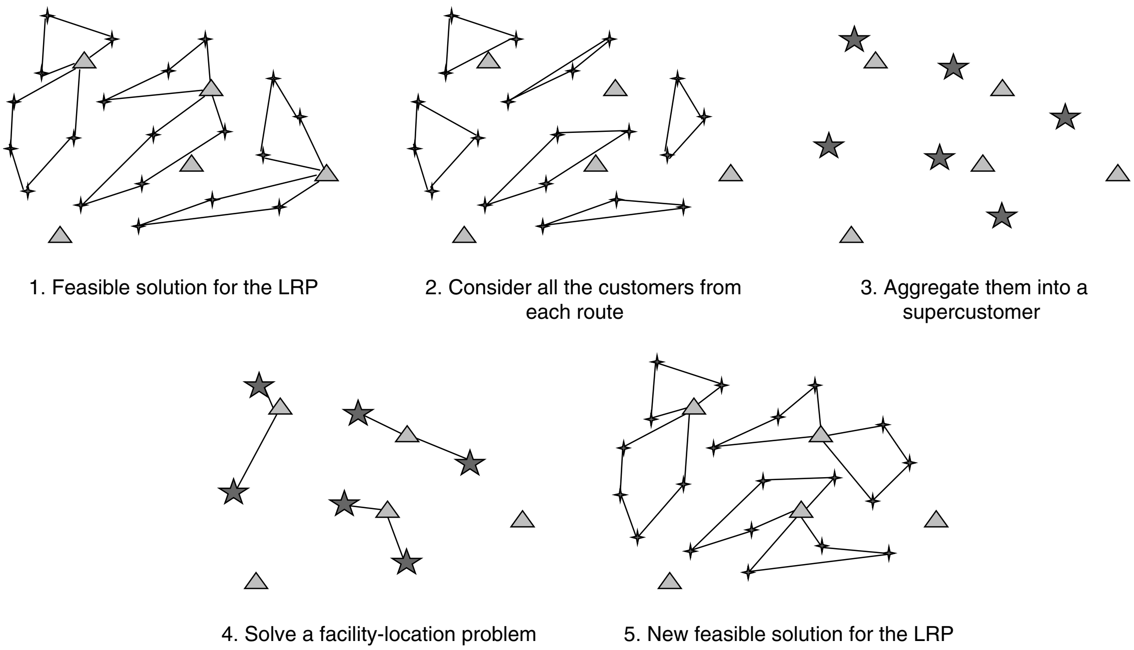

Figure 1 Location Phase

supercustomer k is assigned to depot i:

$$\min \quad Z = \sum_{i\in I} O_i y_i + \sum_{i\in I}\sum_{k\in K} C_{ik} X_{ik} \tag{12}$$

$$\text{subject to} \quad \sum_{i\in I} X_{ik} = 1 \quad \forall\, k \in K \tag{13}$$

$$\sum_{k\in K} D_k X_{ik} \le W_i y_i \quad \forall\, i \in I \tag{14}$$

$$X_{ik} \in \{0, 1\} \quad \forall\, i \in I,\ \forall\, k \in K \tag{15}$$

$$y_i \in \{0, 1\} \quad \forall\, i \in I. \tag{16}$$

In contrast with the LRP model (1)–(11), the fixed costs incurred when a route is opened and the constraints specifically dedicated to the routing—(3)–(4) and (6)–(7)—disappear. Constraints (2) and (5) induce constraints (13), ensuring the assignment of each supercustomer to exactly one depot. Constraints (14) replace constraints (8), ensuring that supercustomers can only be assigned to an open depot that has enough capacity to accommodate them.

2.2. Lagrangean Relaxation

Like Beasley (1993), after comparing different Lagrangean heuristics for the single-source facility-location problem, Ahlander (1994) found that the relaxation proposed by Pirkul (1987) provides the best feasible solutions. Accordingly, our method relaxes the set of constraints concerning the single assignment (constraints (13)). Let u_k, $k \in K$, and $u_k \in \mathbb{R}$ be the Lagrangean multipliers associated with each constraint (13). The relaxed model providing a lower bound for given multipliers is stated as follows:

$$\min \quad g(u) = \sum_{i\in I} O_i y_i + \sum_{i\in I}\sum_{k\in K} C_{ik} X_{ik} + \sum_{k\in K} u_k \left(1 - \sum_{i\in I} X_{ik}\right) \tag{17}$$

$$\text{subject to} \quad \sum_{k\in K} D_k X_{ik} \le W_i y_i \quad \forall\, i \in I \tag{18}$$

$$X_{ik} \in \{0, 1\} \quad \forall\, i \in I,\ \forall\, k \in K \tag{19}$$

$$y_i \in \{0, 1\} \quad \forall\, i \in I. \tag{20}$$

This linear program can be decomposed into m independent subproblems, one for each depot i, formulated as follows:

$$\min \quad \bar{g}_i(u) = O_i y_i + \sum_{k\in K} (C_{ik} - u_k) X_{ik} \tag{21}$$

$$\text{subject to} \quad \sum_{k\in K} D_k X_{ik} \le W_i y_i \tag{22}$$

$$X_{ik} \in \{0, 1\} \quad \forall\, k \in K \tag{23}$$

$$y_i \in \{0, 1\}. \tag{24}$$

If it is assumed that $y_i = 1$, the problem can be seen as m knapsack problems: one for each depot. Here, the m knapsack problems are solved by a classical dynamic programming algorithm (e.g., see Pisinger 1997). Note that the number of depots to open is not to be minimized in the LRP. It may be interesting to open two depots or more, even if it is possible to serve the customers with a single source. Thus, the approach can be adapted to solve instances where depots are uncapacitated by imposing a fictitious capacity equal to half the total demand of the supercustomers when solving the knapsack problems.

For given values of u_k, the optimal solution ($\overline{y}_i$ and $\overline{X}_{ik}$) of (17)–(20), which constitutes a lower bound for (12)–(16), can be deduced from the optimal solutions of the m knapsack problems by considering the values $\bar{g}_i$, which can be seen as the reduced cost for each depot i. If it is negative, then depot i is open ($\overline{y}_i = 1$) and the values of the $\overline{X}_{ik}$, $\forall\, k \in K$ are the ones found by solving the knapsack problem for depot i (X_{ik}). Otherwise, the depot i is closed, and the associated $\overline{y}_i$ and $\overline{X}_{ik}$, $\forall\, k \in K$ are obviously set to zero.

2.3. Subgradient Optimization

The solution obtained with $\overline{y}_i$ and $\overline{X}_{ik}$ only yields a lower bound to the original location problem because it may assign supercustomers to several depots, which is forbidden by the original constraints. To improve this lower bound, the Lagrangean multipliers are adjusted by a subgradient optimization in which the depots that are closest to the supercustomers are favored at the beginning by initializing u_k values to

$$u_k = \min_i (C_{ik}) \quad \forall\, k \in K. \tag{25}$$

Each iteration of the subgradient optimization starts by solving (17)–(20) with the current values of u_k, using the method described in §2.2. The resulting solution is optimal for the original location problem if each supercustomer is assigned to a single depot, which means that it is feasible. Otherwise, we have a lower bound (lb). An upper bound (ub) can be deduced from lb by using the following repair procedure to deal with supercustomers that are split or left unassigned. The knapsack problems deal with capacity constraints, and attention should mainly be paid to unassigned supercustomers. The repair procedure is inspired by the Lagrangean heuristic in Cortinhal and Captivo (2003). Let PD_k be the set of depots to which a supercustomer k is assigned. The first considered supercustomers are the ones assigned to a single depot ($|\mathrm{PD}_k| = 1$). Their assignment is already determined and provides an initial subset of open depots (SOD). Then supercustomers with $|\mathrm{PD}_k| > 1$ are considered. For such a supercustomer k, depots from $\mathrm{SOD} \cap \mathrm{PD}_k$ are considered and the nearest one chosen. The residual capacity of that depot is adjusted accordingly. If none is found, supercustomer k is assigned to the depot in $\mathrm{PD}_k \backslash \{\mathrm{SOD} \cap \mathrm{PD}_k\}$ with the largest number of supercustomers assigned to it in the solution of (17)–(20). This depot is added to SOD. Finally, because of the capacity constraints, the unassigned supercustomers ($|\mathrm{PD}_k| = 0$) are handled one by one in decreasing order of their demand. They are assigned to their nearest depot from SOD with enough capacity. If none can serve the considered supercustomer, then the closed depot that has enough capacity to accommodate it, and for which the total cost of opening plus assignment of the supercustomer is minimal, is opened and added to SOD.

With the lower and upper bounds obtained, the Lagrangean multipliers can be updated using Equation (26). The step length $t^{(q)}$ at iteration q of the subgradient optimization is given by (27). $\overline{ub}$ is the best-known upper bound. $\lambda^{(q)}$ is initially set to 1.5, halved every three consecutive iterations without improvement, and reset to its initial value when it becomes smaller than some predetermined ε or when $\overline{ub}$ is improved. $d^{(q)}$ is an m-vector in which $d_k^{(q)}$ is defined by (28).

$$u_k^{(q)} = u_k^{(q-1)} + t^{(q)} d_k^{(q)} \tag{26}$$

$$t^{(q)} = \lambda^{(q)} \frac{\overline{ub} - \mathrm{lb}}{\|d^{(q)}\|} \tag{27}$$

$$d_k^{(q)} = 1 - \sum_{i \in I} X_{ik}^{\mathrm{lb}} \quad \forall\, k \in K \tag{28}$$

The subgradient optimization continues until convergence occurs ($\mathrm{lb} = \overline{ub}$ or $\|d^{(q)}\| = 0$) as the depots to open as well as the assignment of the supercustomers are directly provided, or a predetermined number of iterations *LagrIt* is reached, in which case the final feasible solution of the location problem is provided by the best upper bound found ($\overline{ub}$).

2.4. From the Facility-Location Problem Solution to the LRP Solution

The solution of the LRP is extracted from the solution of the facility-location problem by disaggregating supercustomers into routes and customers. In fact, the routes available at the beginning of the location phase are kept, but the depots are removed. The new open depots are inserted between two consecutive customers in each route assigned to them, to minimize the insertion cost of these depots. The principle of the location phase is illustrated in Figure 1.

3. Routing Phase with a Granular Tabu Search

The routing phase, based on the GTS defined below, is intended as an intensification phase that optimizes the routes without changing the set of open depots provided by the last location phase, hence resulting in a multidepot VRP. However, all the customers assigned to one depot may be progressively moved to other depots and, ultimately, empty depots may be closed.

Tabu search (Glover and Laguna 1997) is a local search metaheuristic that performs the best available move in a given neighborhood even if it does not improve the current solution, enabling the search to escape from local optima. A tabu mechanism (e.g., a list of recent moves) is used to avoid loops resulting from coming back to already visited solutions. The idea of granularity was introduced by Toth and Vigo (2003). The list of possible moves in the neighborhood is restricted, removing elements that have no real chance of belonging to the optimal solution. This can be seen as an intensification mechanism. Because GTS searches a smaller neighborhood, it is faster than the original TS.

Unlike the GTS proposed by Toth and Vigo (2003), the neighborhood used here consists only of all the solutions that can be obtained by transferring one node from its current position to another one (no exchanges of two, three, or four arcs). It is explored by looking for an unused and nontabu edge (i, j), linking two customers, to insert into the solution. This induces two possible moves in the considered neighborhood, as shown in Figure 2. Three edges must then be deleted from the solution, and two other edges are required to restore feasibility. Once a move is executed, the three deleted edges are considered tabu during a small constant tenure.

A granularity principle is applied to reduce the neighborhood size and prevent the insertion of some nonpromising edges. A reduced graph is considered, in which only edges (i, j) linking customers having a cost smaller than $(\beta\tilde{d})$ are allowed, where β is a constant and $\tilde{d}$ is the average cost of the edges linking two customers in a good feasible solution, as suggested in Toth and Vigo (2003). This will result in the removal of the expensive edges that usually connect two very distant customers and that have very little chance of belonging to an optimal solution. If no improvement occurs after a given number

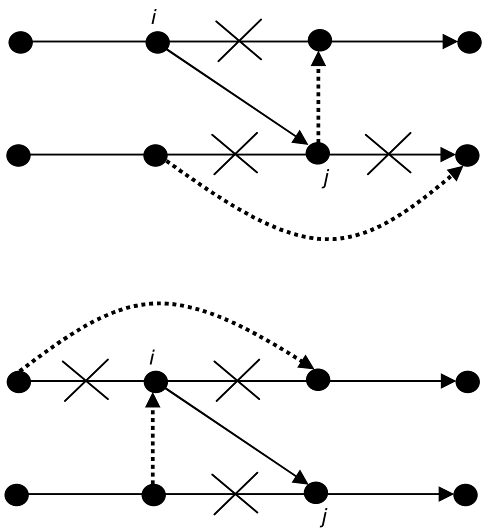

Figure 2 Possible Moves in GTS When Inserting an Edge

of successive iterations (*NoImp*) of the GTS with this initial value of β, the granularity is changed. A less-restricted graph is used by adding edges whose cost is less than $(\beta'\tilde{d})$, with $\beta' > \beta$. This new granularity is kept until the best feasible solution found during the current GTS phase is improved or *Div* iterations have been completed. In both cases, the initial granularity is restored. Such a dynamic adjustment allows the search to be diversified and may lead to a new area of the solution space.

The good feasible solution previously alluded to find $\tilde{d}$ is obtained by executing the randomized constructive heuristic described hereafter, followed by a local search LS explained afterwards. The heuristic randomly selects a depot and uses the simple nearest neighbor algorithm (NNA) to create routes assigned to it until the closest unserviced customer to insert requires more capacity than remains. Then it randomly selects another depot to open. In the case of uncapacitated depots, a number of depots (two or three) is randomly selected and opened with equal probability before building the routes. To initiate a new route, a customer is randomly chosen and assigned to the closest available depot. The other customers on the route are then selected by the NNA. Additional routes are created and filled following these steps until all customers are assigned. The algorithm iterates $2m$ times (to test various sets of open depots), building a new solution each time and keeping the best one found.

Unfortunately, escaping from local optima remains difficult, because the moves are particularly limited by the capacity constraints. To free up the search, capacity violations on both depots and routes are allowed, but subject to a penalty mechanism as follows. Whenever the capacity restriction on a route or a depot is violated by an amount *OverCapa*, a penalty term $P \times OverCapa$ is added to the objective function value, where P is the unit penalty factor. This factor is dynamically adjusted to reflect the relative need to widen the search space or to refocus on strictly feasible solutions. If no infeasible solutions have been visited over the last *NoPenaltyIt* iterations, then the penalty weight is reduced by updating $P = P \times RedPenalty$, where $RedPenalty < 1$. Conversely, if the number of consecutive infeasible solutions visited reaches *PenaltyIt*, or if the number of violations in any visited solution exceeds *MaxPenalty*, then the search is redirected toward feasible solutions by strengthening the penalty through the update $P = P \times IncPenalty$, where $IncPenalty > 1$.

The GTS is performed during *GTSit* iterations. As it explores a restricted neighborhood, the algorithm tries to further improve the solution by performing the local search phase (LS) from Prins, Prodhon, and Wolfler Calvo (2005). Each iteration of this LS explores three neighborhoods, different from the one searched in GTS, in the following given order. In all three cases, capacity constraints on depots and routes must be respected.

• *MOVE*. One customer is shifted from its current position to another position, in the same route or in a different route, which may be assigned to the same depot or not.

• *SWAP*. Two customers are exchanged. They may belong to the same route or, if residual capacities allow it, to two distinct routes sharing one common depot, or not.

• *OPT*. This is an extended 2-opt procedure in which two nonconsecutive edges are removed, either in the same route (this case is equivalent to the well-known 2-opt move; see Lin and Kernighan 1973) or in two distinct routes assigned to a common depot or not. When they belong to different routes, there are several ways of reconnecting the trips. If they are from different depots, edges connecting the last customers of the two considered routes to their depot have to be replaced to satisfy the constraint imposing that a route must begin and finish at the same depot.

The three neighborhoods can be searched in $O(n^2)$. At each iteration, LS performs the first improving move detected. It stops when no additional improvement is reached. As the routing phase does not modify the set of open depots, the overall algorithm goes back to a new phase of location, where the Lagrangean relaxation is applied again.

4. The Whole Cooperative Heuristic

The approach proposed to solve the LRP uses problem-specific knowledge. It deals with a depot location phase, obtained by aggregation of the customers belonging to each route into supercustomers and solved by Lagrangean relaxation (LR), and a routing phase, handled by a GTS. The method is called LRGTS. A natural cooperation steps in with no hierarchy. This synergy strengthens the method: The location phase provides a fixed set of open depots, and the routing phase gives the supercustomers from the aggregation of the customers of each route. A global iteration of the algorithm corresponds to solving both phases. However, the basic scheme of the method converges too fast to a local optimum. Indeed, depot location strongly depends on the choice of the supercustomers, and the routing (from which the supercustomers come) is strongly determined by the depot location. Thus, a cooperative restart has to be introduced to provide further information and free up the search simultaneously.

At the end of each global iteration, a frequency counter is updated for each edge used and the convergence of the algorithm is checked. If the current

//Initialization
Create a good solution S_0 for GTS with a heuristic
Create the granular graphs by using β and β'
NbRestart $= 0$ *//Number of restarts executed*
$C_{best} = Cost(S_0)$
  //Initialize the cost of the best solution with the cost of S_0
$S_{best} = S_0$ *//Initialize the best solution with* S_0
$C_{last} = +\infty$ *//Initialize the cost of the last global iteration*
Create the initial solution S of the cooperative method
//Main Loop
repeat
  //Restart
  if $(Cost(S) \geq C_{last})$ **then**
    Create new routes using edge frequencies
    $LastCost = +\infty$
    $NbRestart = NbRestart + 1$
  else
    $C_{last} = Cost(S)$
  end if
  //Location phase
  Create the supercustomers from S
  Operate the location step with Lagrangean relaxation
  Convert the supercustomers into routes to have a feasible LRP solution S
  //Routing phase
  Apply GTS to S
  Apply LS to S
  //Memorization
  Update edge frequency for each edge used in S
  if $(Cost(S) < C_{best})$ **then**
    $C_{best} = Cost(S)$
    $S_{best} = S$
  end if
until $(NbRestart > MaxRest)$
return(S_{best})

Figure 3 **Overview of the Algorithm**

solution has not been improved during the global iteration, a restart is applied and takes place before the beginning of the location phase (see Figure 3). The trips resulting from the last routing phase are discarded and rebuilt by a heuristic in which the number of edges Δ to connect customers is limited. The goal is to give some flexibility to the location phase by computing new trips (more numerous) that, once aggregated, give super customers with smaller demand. The heuristic builds a list L of edges sorted in decreasing frequency order and initializes U, the list of edges linking customers inserted into the trips, to an empty set. Then, iteratively, the first elements of L (beginning with the most frequently used edges) are added to U, provided that no fork is created (the degree of each customer node is at most two) and that the load of each connected component of U does not exceed the vehicle capacity. It stops either when $|U| = \Delta$ or when no connected component can be extended without violating vehicle capacity. The resulting connected components in U provide the supercustomers for the following location phase. However, to keep some diversity at each restart, each edge in L is considered with a given probability *prob*.

Δ is initially set to a small value Δ_{init} in relation with the maximum number of edges in a solution. Afterward, at each new restart, as the information provided by edge frequency counters becomes more and more accurate, Δ increases by a factor $g > 1$. This principle can be seen as fixing an increasing number of variables in each new solution. At the beginning of the algorithm, no edge frequency is available, so the initial solution of the algorithm is built with NNA limiting the load of routes to the maximal customer demand.

An overview of the method is given in Figure 3. The parameter *MaxRest* represents the number of restarts to perform used as the stopping criterion. With the diversification technique exploiting information on the most promising edges linking customers, the algorithm becomes more efficient and provides better solutions, as shown in the next section.

5. Computational Study

LRGTS is evaluated on three sets of randomly generated Euclidean instances described in §5.1. The setting of the parameters, the three algorithms and the lower bound used for comparisons are briefly presented in §5.2. Tables of results are provided and discussed in §5.3. Subsection 5.4 provides further analysis about the obtained solutions.

5.1. Instances

The first set contains 30 LRP instances with capacitated routes and depots, created for our previous work on a multistart local search method (Prins, Prodhon, and Wolfler Calvo 2004) and also used for a GRASP (Prins, Prodhon, and Wolfler Calvo 2005). It contains the largest instances with capacitated depots (200 customers) and is reused in this study to compare the performances of LRGTS with respect to the best-known results (BKR) and GRASP. Its main characteristics are shown in Table 1: number of depots $m \in \{5, 10\}$, number of clients $n \in \{20, 50, 100, 200\}$, vehicle capacity $Q \in \{70, 150\}$, and number of clusters $\beta \in \{0, 2, 3\}$. The case $\beta = 0$ corresponds in fact to a uniform distribution of customers in the Euclidean plane. These instances in which all numbers are integer were generated as follows. For given choices of m, n, Q, and β, the customers' locations are randomly

Table 1 **Results for the First Set with Capacitated Depots, with Gap to BKR**

				BKR	GRASP					LRGTS				
n	m	β	Q	Cost	Cost	dep	rt	T	Gap	Cost	dep	rt	T	Gap
20	5	0	a	54,793	**55,021**	3	5	0.2	0.42	55,131	3	5	0.4	0.62
20	5	0	b	39,104	**39,104***	2	3	0.2	0.00	**39,104***	2	3	0.2	0.00
20	5	2	a	48,908	**48,908***	3	5	0.1	0.00	**48,908***	3	5	0.5	0.00
20	5	2	b	37,542	**37,542***	2	3	0.2	0.00	**37,542***	2	3	0.1	0.00
50	5	0	a	90,111	90,632	3	12	1.8	0.58	**90,160**	3	12	0.3	0.05
50	5	0	b	63,242	64,761	2	6	1.8	2.40	**63,256**	2	6	1.0	0.02
50	5	2	a	88,298	88,786	3	12	2.4	0.55	**88,715**	3	12	1.8	0.47
50	5	2	b	67,340	68,042	3	6	2.5	1.04	**67,698**	3	6	1.8	0.53
50	5	2'	a	84,055	**84,055***	3	12	1.7	0.00	84,181	3	12	2.0	0.15
50	5	2'	b	51,822	52,059	3	6	2.6	0.46	**51,992**	3	6	0.9	0.33
50	5	3	a	86,203	87,380	2	12	2.3	1.37	**86,203***	2	12	0.3	0.00
50	5	3	b	61,830	61,890	2	6	2.0	0.10	**61,830***	2	6	0.5	0.00
100	5	0	a	275,993	279,437	3	24	27.6	1.25	**277,935**	3	24	8.7	0.70
100	5	0	b	214,392	216,159	3	12	23.2	0.82	**214,885**	3	11	8.3	0.23
100	5	2	a	194,598	199,520	2	24	17.4	2.53	**196,545**	2	24	2.3	1.00
100	5	2	b	157,173	159,550	2	11	22.4	1.51	**157,792**	2	11	3.3	0.39
100	5	3	a	200,246	203,999	2	25	21.6	1.87	**201,952**	2	24	2.4	0.85
100	5	3	b	152,586	**154,596**	2	11	20.3	1.32	154,709	2	12	2.9	1.39
100	10	0	a	290,429	323,171	4	26	37.4	11.27	**291,887**	3	26	14.1	0.50
100	10	0	b	234,641	271,477	4	12	29.5	15.70	**235,532**	3	12	14.0	0.38
100	10	2	a	244,265	254,087	3	25	39.1	4.02	**246,708**	3	24	14.4	1.00
100	10	2	b	203,988	206,555	3	11	29.8	1.26	**204,435**	3	11	10.1	0.22
100	10	3	a	253,344	270,826	3	25	35.4	6.90	**258,656**	3	25	13.3	2.10
100	10	3	b	204,597	216,173	3	11	39.8	5.66	**205,883**	3	11	10.8	0.63
200	10	0	a	479,425	490,820	3	48	517.5	2.38	**481,676**	3	47	62.0	0.47
200	10	0	b	378,773	416,753	3	22	379.1	10.03	**380,613**	3	22	60.3	0.49
200	10	2	a	450,468	512,679	3	49	554.3	13.81	**453,353**	3	48	60.3	0.64
200	10	2	b	374,435	379,980	3	23	367.4	1.48	**377,351**	3	23	76.9	0.78
200	10	3	a	472,898	496,694	3	46	424.8	5.03	**476,684**	3	47	77.2	0.80
200	10	3	b	364,178	389,016	3	22	290.2	6.82	**365,250**	3	22	73.3	0.29
Average				197,323	207,322	2.8	17.2	96.5	**3.35**	198,552	2.7	17.1	17.5	**0.50**

*Solution cost is equal to the BKR.

Table 2 Results for the Second Set with Capacitated Depots, with Gap to LB

			LB	CH		GRASP				LRGTS			
n	*m*	*Q*	Cost	Cost	Gap	Cost	*dep*	*rt*	Gap	Cost	*dep*	*rt*	Gap
21	5	6,000	**424.9***	435.9	2.6	429.6	2	5	1.1	**424.9***	2	4	0.0
22	5	4,500	**585.1***	591.5	1.1	**585.1***	1	3	0.0	587.4	1	3	0.4
27	5	2,500	**3,062.0***	**3,062.0***	0.0	**3,062.0***	2	4	0.0	3,065.2	2	4	0.1
29	5	4,500	**512.1***	**512.1***	0.0	515.1	2	4	0.6	**512.1***	2	4	0.0
32	5	8,000	556.5	**571.7**	2.7	571.9	2	4	2.8	584.6	2	4	5.0
32	5	11,000	**504.3***	511.4	1.4	**504.3***	1	3	0.0	504.8	1	3	0.1
36	5	250	**460.4***	470.7	2.2	**460.4***	1	4	0.0	476.5	1	4	3.5
50	5	160	549.4	**582.7**	6.1	599.1	3	8	9.1	586.4	2	6	6.7
75	10	140	744.7	886.3	19.0	**861.6**	3	9	15.7	863.5	3	10	16.0
88	8	9,000,000	356.4	384.9	8.0	**356.9**	2	8	0.1	368.7	2	6	3.5
100	10	200	788.6	889.4	12.8	861.6	3	9	9.3	**842.9**	2	8	6.9
134	8	850	—	6,238.0	7.4	5,965.1	4	11	2.7	**5,809.0**	3	11	0.0
150	10	8,000,000	43,406.0	46,642.7	7.5	44,625.2	3	13	2.8	**44,386.3**	3	14	2.3
Average			4,443.0	4,752.3	**5.4**	4,569.1	2.2	6.5	**3.4**	4,539.4	2.0	6.2	**3.4**

*Proven optima.

chosen in the Euclidean plane, and the traveling costs c_{ij} correspond to the distances, multiplied by 100 and rounded up to the nearest integer. The two instances with 50 customers, $\beta = 2'$, contain two strongly separated clusters. Finally, the other data (demands, depot capacities, and fixed costs) are selected. In particular, each demand follows a uniform distribution in interval {10, 20} and depot capacities are chosen in such a way that at least two or three depots must be opened.

The second set of 13 instances listed in Table 2 was gathered by Barreto (2004b) in his thesis on clustering heuristics for the LRP (see Barreto 2004a for site to download). These files either come from the LRP literature or are obtained by adding several depots to classical VRP instances. Their traveling costs are equal to the Euclidean distances (not rounded), and all routes are capacitated. Except in a few instances, all depots also have a limited capacity. The table shows that vehicle capacity and demands are sometimes huge. The main interest of these instances resides in the lower bounds given by Barreto and in the solution values obtained by his heuristics. No other set enables an estimate of the deviation of algorithms to optimality.

The third set comprises 36 instances with uncapacitated depots designed by Tuzun and Burke (1999) to evaluate a TS. They use $n \in \{100, 150, 200\}$, $m \in \{10, 20\}$, $Q = 150$, and uniform demands in interval {1, 20}. As in the first set, spatial distribution of customers is controlled: There are zero, three, or five clusters (zero corresponding to a uniform distribution), and the percentage of customers P that belong to a cluster is fixed at 75% or 100%. As in the second set, distances are not rounded.

5.2. Implementation, Parameters, and Algorithms Compared

Our earlier algorithm GRASP and the LRGTS are coded in Visual C++ and have been tested on a Dell PC Optiplex GX260, with a 2.4 GHz Pentium 4, 512 MB of RAM, and Windows XP. The following parameters for LRGTS have been selected after a preliminary testing phase in order to provide the best average solution values:

- $\Delta_{init} = 3 + (n - \hat{k}) \times 0.25$ with $\hat{k} = \sum_{j \in J} d_j / Q$, maximal number of restarts $MaxRest = n/3$;
- location phase: $g = 1.15$, $prob = 0.80$, $LagrIt = 200$;
- GTS phase: $GTSit = n$, $\beta = 1.20$, $\beta' = 1.85$, $NoImp = 0.05n$, $Div = 0.20n$, initial value of $P = 2\beta\tilde{d}/AvDem$ (*AvDem* being the average customer demand), $NoPenaltyIt = 3$, $MaxPenalty = 3$, $PenaltyIt = 0.08n$, $RedPenalty = 0.8$, $IncPenalty = 1.4$, and a tabu tenure equal to $0.15n$.

The GRASP described in Prins, Prodhon, and Wolfler Calvo (2005) is based on an extended and randomized version of the Clarke and Wright savings heuristic for the VRP: When two routes are merged, the new route may be reassigned to another depot, which can be closed or already open. The purpose of this heuristic is to provide good trial solutions, with a fixed set of open depots, to the local search phase. The results are improved by adding a memory to diversify the subsets of open depots and a postoptimization based on a path-relinking process.

The source codes of the other two algorithms (Tuzun and Burke 1999; Barreto 2004b) and of the lower bound presented in the sequel were not available to us. Hence, our tables report the solution values and running times given by their authors.

Barreto's clustering heuristic (2004b) is in fact a generic three-phase method that (1) builds clusters of customers compatible with vehicle capacity, (2) solves a TSP in each cluster, and (3) selects the open depots and assigns the resulting TSP tours to them by solving a single-source capacitated facility-location problem. Twenty-four versions are obtained by combining four clustering methods and six proximity measures, and the best solution is reported for each instance. Barreto has also obtained a lower bound (LB) by applying a cutting-plane algorithm to a relaxed two-index integer linear programming formulation of the LRP. The machine used is not specified.

The last method compared with LRGTS is a TS designed specifically by Tuzun and Burke for the case of uncapacitated depots (Tuzun and Burke 1999). As a result of this assumption, these authors developed a very fast algorithm that solves a sequence of subproblems in which only $1, 2, 3\ldots$ depots are open. This process stops when adding one more depot would result in a degraded solution, and in general this happens early. Apart from the limited number of subproblems that must actually be solved in practice, the local search for each subproblem is very fast, because the moves may exchange the open/closed status of two depots, but without changing the number of open depots that is fixed in each subproblem. Moreover, it is important to note that this fast algorithm *is not applicable* to the first two sets of instances with capacitated depots. Tuzun and Burke report their results on a 266 MHz Pentium II PC (18 times slower than our computer, according to Dongarra 2006).

5.3. Discussion of Results

In the following tables, for each instance, costs in boldface indicate which method obtains the best result, and times T are given in seconds. The costs reported as BKR are the best-known results. They have been obtained during testing phases of our methods, including LRGTS, using various parameters. The gaps between each method and a bound taken as reference (either BKR—Tables 1 and 3—or the lower bound—Table 2) are given in percentage. The numbers of depots and routes opened in the solution are reported, respectively, in columns *dep* and *rt*.

5.3.1. First Set. Table 1 provides a comparison between LRGTS and our earlier method GRASP with the BKR on the 30 instances from the first set. Asterisks mean that solution cost is equal to the BKR.

Thanks to the granularity system (see §5.4), LRGTS is by far the fastest method, nearly 5.5 times faster than GRASP, while finding a best solution for 26 of the 30 instances and reaching 5 times the BKR. Its better average solution quality can be explained by its robustness, which results from the use of specific knowledge about the problem, while the GRASP is a generic method with independent iterations. However, GRASP is as good as LRGTS on the smallest instances ($n = 20$), but its deviation to BKR exceeds 10% on four large instances ($n = 100$ and 200).

5.3.2. Second Set. In Table 2, LRGTS is compared with GRASP and with the clustering heuristics CH and lower bound LB from Barreto. Asterisks indicate proven optima. The *Gap* columns correspond to deviations to LB in percentage, except for the next-to-last instance ($n = 134$ and $m = 8$) without lower bound, where they give deviations to the best method (LRGTS).

Both LRGTS and GRASP outperform Barreto's cocktail of 24 clustering heuristics and achieve almost the same costs on average (gap around 3%). This confirms the observation done on the first set: GRASP seems to work better on small instances, frequent in the second set, while LRGTS is more efficient on large ones.

The comparison with LB does not permit a conclusion about the quality of the obtained solutions, except on small instances, when the lower bound is proven optimal. On some large instances, the deviation may exceed 6% for all heuristics. Nevertheless, it is difficult to know whether the optimum is closer to the LB or to the best heuristic.

Running times are not provided for each instance because of lack of space. In fact, all approximate methods require less than one second for up to 50 customers. For $n > 100$, each clustering heuristic needs nearly one second (but 24 such heuristics are executed). When $n < 100$, LRGTS runs in less than six seconds and GRASP in less than 24 seconds. On the two last (and largest) instances, LRGTS needs 13 and 31 seconds, respectively, versus 50 and 156 seconds for GRASP. Therefore, the GRASP gives slightly better results on average, but at the expense of a computational effort quadrupled, compared with LRGTS.

5.3.3. Third Set. Table 3 compares LRGTS with GRASP and with Tuzun and Burke's TS. LRGTS appears to be the most efficient method (gap of 1.32% with BKR, while TS and GRASP achieve 3.79% and 2.97%, respectively); that is remarkable because it is not tailored for the special case of uncapacitated depots. Of course, TS is much faster (reported times are not scaled), but remember that it applies a strategy (progressive incrementation of the number of open depots) and a local search that are no longer valid for the capacitated case. Furthermore, TS has an average gap more than two times higher than the one obtained using LRGTS, and it never provides the best cost. More precisely, its gap to BKR is always above 1%. Anyway, the larger duration of LRGTS is still reasonable for a strategic optimization problem that does not need to be solved every day. It is also interesting to notice that VRP literature reports much larger running times for VRP instances of 200 customers.

Table 3 Results for the Third Set with Uncapacitated Depots, with Gap to BKR

				BKR	*TS*			GRASP					LRGTS				
n	*m*	*β*	*P*	Cost	Cost	*T*	Gap	Cost	*dep*	*rt*	*T*	Gap	Cost	*dep*	*rt*	T	Gap
100	10	0	0.75	1,468.40	1,556.64	5	6.01	1,525.25	3	11	32.4	3.87	**1,490.82**	3	11	3.3	1.53
100	20	0	0.75	1,449.20	1,531.88	3	5.71	1,526.90	3	11	40.7	5.36	**1,471.76**	3	11	6.5	1.56
100	10	0	1	1,396.46	1,443.43	3	3.36	1,423.54	2	11	27.6	1.94	**1,412.04**	3	10	4.2	1.12
100	20	0	1	1,432.29	1,511.39	4	5.52	1,482.29	2	11	36.2	3.49	**1,443.06**	2	11	7.4	0.75
100	10	3	0.75	1,167.53	1,231.11	4	5.45	1,200.24	2	11	27.7	2.80	**1,187.63**	2	11	6.9	1.72
100	20	3	0.75	1,102.70	1,132.02	2	2.66	1,123.64	3	11	34.3	1.90	**1,115.95**	3	11	6.8	1.20
100	10	3	1	793.97	825.12	3	3.92	814.00	3	12	22.5	2.52	**813.28**	3	12	5.2	2.43
100	20	3	1	728.30	**740.64**	3	1.68	747.84	3	11	37.3	2.68	742.96	3	11	5.9	2.01
100	10	5	0.75	1,238.49	1,316.98	3	6.34	1,273.10	3	11	21.5	2.79	**1,267.93**	3	11	4.3	2.38
100	20	5	0.75	1,246.34	1,274.50	4	2.26	1,272.94	2	11	36.0	2.13	**1,256.12**	3	11	6.3	0.78
100	10	5	1	902.38	920.75	4	2.04	**912.19**	3	12	20.3	1.09	913.06	3	12	4.0	1.18
100	20	5	1	1,018.58	1,042.21	3	2.32	**1,022.51**	3	11	38.4	0.39	1,025.51	3	11	4.9	0.68
150	10	0	0.75	1,866.75	2,000.97	12	7.19	2,006.70	3	16	113.0	7.50	**1,946.01**	3	16	12.5	4.25
150	20	0	0.75	1,841.86	1,892.84	12	2.77	1,888.90	4	16	161.4	2.55	**1,875.79**	3	16	18.5	1.84
150	10	0	1	1,981.37	2,022.11	14	2.06	2,033.93	3	17	100.0	2.65	**2,010.53**	3	16	11.1	1.47
150	20	0	1	1,809.25	1,854.97	13	2.53	1,856.07	4	16	132.4	2.59	**1,819.89**	3	16	15.8	0.59
150	10	3	0.75	1,448.27	1,555.82	9	7.43	1,508.33	3	16	117.7	4.15	**1,448.65**	2	16	22.0	0.03
150	20	3	0.75	1,444.25	1,478.80	12	2.39	**1,456.82**	2	16	166.1	0.87	1,492.86	3	16	28.0	3.37
150	10	3	1	1,206.73	1,231.34	9	2.04	1,240.40	2	16	106.7	2.79	**1,211.07**	3	17	14.6	0.36
150	20	3	1	931.94	948.28	9	1.75	940.80	3	17	142.4	0.95	**936.93**	3	17	13.7	0.54
150	10	5	0.75	1,699.92	1,762.45	9	3.68	1,736.90	3	17	92.8	2.18	**1,729.31**	3	16	17.9	1.73
150	20	5	0.75	1,401.82	1,488.34	9	6.17	1,425.74	3	16	128.4	1.71	**1,424.59**	3	16	18.5	1.62
150	10	5	1	1,199.51	1,264.63	10	5.43	1,223.70	3	17	88.5	2.02	**1,216.32**	3	16	14.5	1.40
150	20	5	1	1,152.86	1,182.28	9	2.55	1,231.33	4	17	134.9	6.81	**1,162.16**	3	17	14.3	0.81
200	10	0	0.75	2,259.87	2,379.47	22	5.29	2,384.01	3	21	308.0	5.49	**2,296.52**	3	22	32.6	1.62
200	20	0	0.75	2,185.55	2,211.74	22	1.20	2,288.09	4	22	410.0	4.69	**2,207.50**	4	22	39.6	1.00
200	10	0	1	2,234.78	2,288.17	23	2.39	2,273.19	3	21	311.4	1.72	**2,260.87**	4	21	32.8	1.17
200	20	0	1	2,250.34	2,355.81	26	4.69	2,345.10	3	22	418.9	4.21	**2,259.52**	3	21	40.2	0.41
200	10	3	0.75	2,101.90	2,158.60	20	2.70	2,137.08	3	22	338.0	1.67	**2,120.76**	3	21	47.2	0.90
200	20	3	0.75	1,709.56	1,787.02	18	4.53	1,807.29	4	21	370.0	5.72	**1,737.81**	3	21	59.3	1.65
200	10	3	1	1,467.54	1,549.79	18	5.60	1,496.75	2	21	242.7	1.99	**1,488.55**	2	21	36.7	1.43
200	20	3	1	1,084.78	1,112.96	18	2.60	1,095.92	3	22	308.5	1.03	**1,090.59**	3	22	38.7	0.54
200	10	5	0.75	1,973.28	2,056.11	23	4.20	2,044.66	4	23	282.8	3.62	**1,984.06**	4	22	41.6	0.55
200	20	5	0.75	1,957.23	2,002.42	20	2.31	2,090.95	4	22	399.2	6.83	**1,986.49**	4	22	51.8	1.50
200	10	5	1	1,771.06	1,877.30	20	6.00	1,788.70	2	21	199.0	1.00	**1,786.79**	3	22	34.0	0.89
200	20	5	1	1,393.62	1,414.83	17	1.52	1,408.63	5	22	296.3	1.08	**1,401.16**	5	22	43.2	0.54
Average				1,508.85	1,566.77	11.5	**3.79**	1,556.51	3.0	16.4	159.6	**2.97**	1,528.75	3.1	16.3	21.2	**1.32**

5.4. Further Analysis of the Results

The previous discussion shows that LRGTS finds very good solutions in a reasonable amount of time. The aim of this subsection is to confirm the positive roles of granularity and of the additional cooperation. Note that in Tables 5, 6, and 7, LRGTS is taken as a reference for calculating the gap.

5.4.1. Impact of Granularity. Granularity, like any neighborhood reduction technique, may save time at the expense of solution quality. To evaluate this risk, two algorithms are evaluated for the routing phase: the GTS of §3 and the same method, but without granularity, TS. The results are given in Table 4. For each group of instances, this table provides the average time spent in each routing algorithm and the cost variation in percent between the solution submitted to the routing phase and the improved solution returned at the end of GTS or TS (gap). The results indicate that the routing phase duration is divided by 30 when granularity is used, the price to pay being a slightly smaller cost improvement (−5.3% versus −6.5% without granularity).

Surprisingly, both the running time and the solution value are improved (on average) if the complete algorithms (location phase plus routing phase) are compared, as in Table 5. LRTS denotes the cooperative method in which GTS is replaced by TS in the routing phase. The values reported for each group of instances are the average cost and running time (for each algorithm) and the saving in percent (gap) brought by granularity.

On average, the total running time is divided by seven, and the solution values are 0.5% better. A possible explanation could be a premature convergence in LRTS, with the full exploration of the neighborhoods in the routing phase. In this case, the algorithm alternates less between location and routing phases

Table 4 Impact of Granularity on Routing

n	*m*	*GTS*	*T*	*TS′*	*T*
First set					
20	5	−9.1	0.0	−10.4	0.0
50	5	−7.8	0.1	−8.8	0.9
100	5	−4.1	0.8	−4.7	14.8
100	10	−2.6	1.1	−3.3	23.2
200	10	−1.6	6.0	−2.5	316.5
Average		−5.0	1.6	−5.9	71.1
Second set					
≤50	5	−8.5	0.0	−10.1	0.2
≤150	≤10	−5.7	2.5	−7.1	41.8
Average		−7.4	1.0	−9.0	16.2
Third set					
100	10	−6.1	1.3	−7.4	22.6
100	20	−5.6	1.2	−6.6	16.0
150	10	−5.0	5.3	−6.2	97.4
150	20	−4.4	4.3	−5.6	100.1
200	10	−3.9	10.8	−5.2	354.1
200	20	−4.3	13.5	−5.7	412.0
Average		−4.9	6.1	−6.1	167.0
Average		**−5.3**	**3.5**	**−6.5**	**105.8**

before performing a restart, and the solution space is less explored.

5.4.2. Impact of Restarts. In addition to the natural cooperation realized by the alternation between the two phases, the restarts based on statistics about the most used edges are expected to diversify the search.

Table 6 shows what happens when restarts are suppressed, using the same indicators as in Table 5: the resulting algorithm is more than five times faster, but the cost increases by 3.6% on average, thus proving the importance of diversifying the search.

In Table 7, which still uses the same indicators, LRGTS is compared with a version using restarts, but without the additional cooperation based on edge statistics. Each restart is based on a heuristic that creates a new route by randomly selecting one customer and completing the route using the NNA. This blind diversification is sufficient to improve the results of the version without restarts (compare with Table 6),

Table 6 Impact of Restarts

n	*m*	LRGTS	*T*	No coop	*T*	Gap
First set						
20	5	45,171	0.3	45,171	0.4	0.0
50	5	74,254	1.1	74,445	1.1	0.3
100	5	200,636	4.6	200,922	4.2	0.1
100	10	240,517	12.8	240,539	10.8	0.0
200	10	422,488	68.3	423,421	56.2	0.2
Average			17.5		14.6	0.1
Second set						
≤50	5	842.74	0.1	849.27	0.1	1.3
≤150	≤10	10,454.0	10.6	10,378.31	9.2	0.5
Average			4.2		3.6	1.0
Third set						
100	10	1,180.79	4.6	1,190.35	4.3	0.8
100	20	1,175.89	6.3	1,183.38	6.5	0.5
150	10	1,593.65	15.4	1,594.86	12.8	0.1
150	20	1,452.03	18.1	1,452.65	16.5	0.1
200	10	1,989.59	37.5	1,991.32	31.9	0.1
200	20	1,780.51	45.5	1,786.44	35.2	0.3
Average			21.2		17.9	0.3
Average			**17.0**		**14.3**	**0.4**

Table 5 Impact of Granularity on Global Solution

n	*m*	LRGTS	*T*	LRTS	*T*	GAP
First set						
20	5	45,171	0.3	45,428	0.3	0.5
50	5	74,254	1.1	75,264	2.1	1.5
100	5	200,636	4.6	200,949	18.1	0.1
100	10	240,517	12.8	242,350	35.5	0.8
200	10	422,488	68.3	421,862	373.2	−0.2
Average			17.5		85.9	0.6
Second set						
≤50	5	842.7	0.1	850.34	0.3	1.5
≤150	≤10	10,454.1	10.6	10,505.08	48.4	1.1
Average			4.2		18.8	1.4
Third set						
100	10	1,180.79	4.6	1,190.38	25.3	0.7
100	20	1,175.89	6.3	1,190.31	20.4	1.1
150	10	1,593.65	15.4	1,602.11	104.5	0.6
150	20	1,452.03	18.1	1,447.82	111.6	−0.3
200	10	1,989.59	37.5	1,980.62	373.2	−0.4
200	20	1,780.51	45.5	1,767.34	435.0	−0.7
Average			21.2		178.3	0.2
Average			**17.0**		**117.0**	**0.5**

Table 7 Impact of Using Information on the Edges

n	*m*	LRGTS	*T*	No coop	*T*	Gap
First set						
20	5	45,171	0.3	45,171	0.4	0.0
50	5	74,254	1.1	74,445	1.1	0.3
100	5	200,636	4.6	200,922	4.2	0.1
100	10	240,517	12.8	240,539	10.8	0.0
200	10	422,488	68.3	423,421	56.2	0.2
Average			17.5		14.6	0.1
Second set						
≤50	5	842.74	0.1	849.27	0.1	1.3
≤150	≤10	10,454.0	10.6	10,378.31	9.2	0.5
Average			4.2		3.6	1.0
Third set						
100	10	1,180.79	4.6	1,190.35	4.3	0.8
100	20	1,175.89	6.3	1,183.38	6.5	0.5
150	10	1,593.65	15.4	1,594.86	12.8	0.1
150	20	1,452.03	18.1	1,452.65	16.5	0.1
200	10	1,989.59	37.5	1,991.32	31.9	0.1
200	20	1,780.51	45.5	1,786.44	35.2	0.3
Average			21.2		17.9	0.3
Average			**17.0**		**14.3**	**0.4**

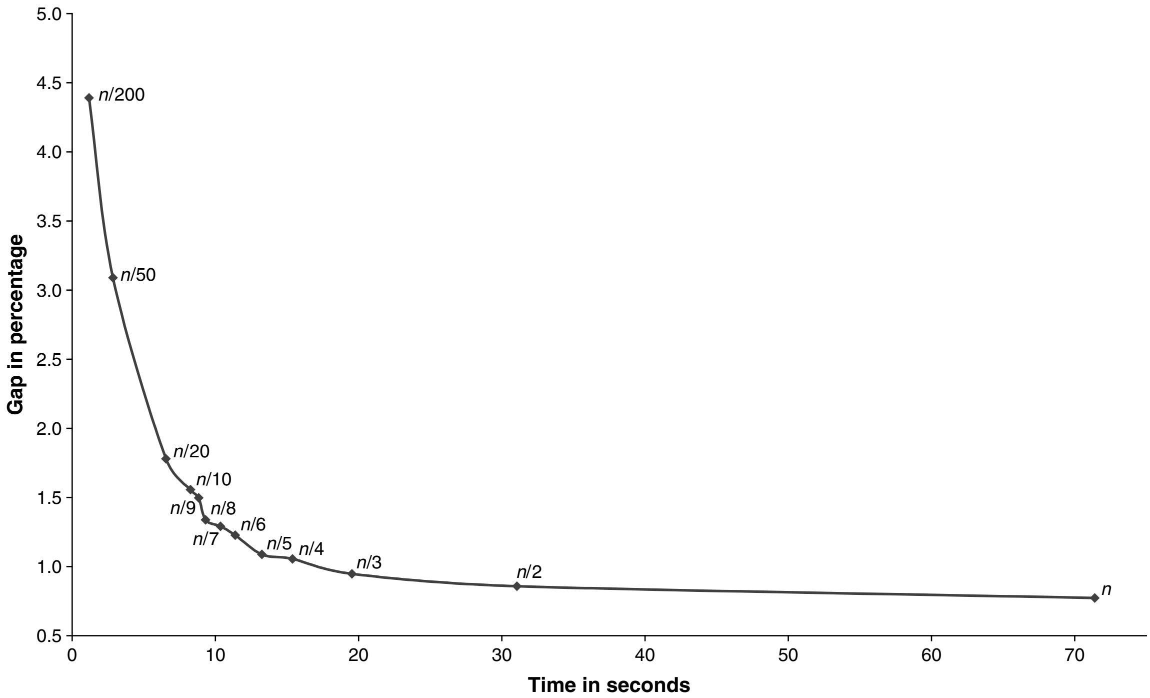

Figure 4 **Impact of Number of Restarts (*MaxRest*)**

but our LRGTS saves 0.4% more. This saving, which seems moderate, must be compared with the average deviation to the bounds.

Finally, Figure 4 gives a more general view of the incidence of the number of restarts used. One point of the graph synthesizes the average performances of the method on instances from the first and third set for a given value of *MaxRest*. On the x-axis is reported the average central processing unit (CPU) time needed to run the algorithm. The y-axis shows the relative gap between the solution cost obtained and BKR. Instances from the second set are not included in the average results because, on these instances, the gap is computed to a lower bound and the reference used for the gap would be different (LB instead of BKR). The range of *MaxRest* is from $n/200$ to n.

The figure illustrates the fact that the method needs some diversifications, brought here by cooperative restarts, to improve the quality of the solution. However, a trade-off has to be found because the CPU time also depends on this parameter, which is the stopping criterion. In the proposed method, *MaxRest* is chosen equal to $n/3$, because the graph shows that this point leads to a gap smaller than 1% in a reasonable amount of time (19.5 seconds on average). The algorithm can obtain a slightly better result by using $MaxRest = n$, for example, but becomes 3.5 times slower. It is also possible to be faster without impairing the solution cost too much. For example, with $MaxRest = n/8$, the CPU time is divided by two, while the costs increase by 0.4%.

Note that in any case the proposed algorithm is much faster than the other metaheuristics, except TS, which is designed for uncapacitated depots and can use a simple and fast search strategy not applicable to LRP with depot capacities. Nevertheless, by reducing the number of iterations, LRGTS can be as fast as TS and find solutions of the same quality (*MaxRest* between $n/50$ and $n/200$ in Figure 4).

6. Conclusion

In this paper, a new metaheuristic for the LRP with both capacitated depots and vehicles is presented. The method is a cooperative metaheuristic (LRGTS) exchanging information between two main phases. The first one is a location phase solved by a facility-location problem that aggregates the routes into supercustomers and uses a Lagrangean relaxation on assignment constraints to select the depot location. The second one is a routing phase dealt with by a GTS. The new routes are used to a new location phase. The cooperation between the phases strengthens the method, but can lead to a premature convergence. Therefore, further cooperation occurs that builds new routes using information about the most promising edges linking customers.

The method is tested on three sets of small-, medium-, and large-scale instances with up to 200 customers. The results show that the proposed LRGTS is able to improve more than 80% of the values obtained either by heuristics or metaheuristics. This is also true when depots are uncapacitated, although it is compared to a TS specially tailored for this particular case of the LRP. Furthermore, even if its good performance is especially noticeable on large scale instances, LRGTS is able to find optimal solutions on small instances by reaching the lower bound computed by Barreto (2004b). These good results can be explained by the use of problem-specific knowledge that in addition leads to a fast algorithm. Moreover, the CPU times can be modulated at the price of a slight variation of the solution costs.

Acknowledgments

The authors would like to thank the referees for their helpful and careful remarks, which have helped improve the quality of the paper. This research is partially supported by the Conseil Regional de Champagne-Ardenne (Champagne-Ardenne District Council) and by the European Social Fund.

References

Ahlander, F. 1994. The capacitated facility location problem with single sourcing. Unpublished doctoral dissertation, Linkoping University, Sweden.

Albareda-Sambola, M., J. A. Díaz, E. Fernández. 2005. A compact model and tight bounds for a combined location-routing problem. *Comput. Oper. Res.* **32**(3) 407–428.

Barreto, S. S. 2004a. Home page of Sérgio Sautos Barreto. http://sweet.ua.pt/˜iscf143.

Barreto, S. S. 2004b. Análise e Modelização de Problemas de localização-distribuição [Analysis and modelling of location-routing problems]. Unpublished doctoral dissertation, University of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal.

Beasley, J. E. 1993. Lagrangean heuristics for location problems. *Eur. J. Oper. Res.* **65** 383–399.

Bruns, A., A. Klose. 1996. A "locate first-route second" heuristic for a combined location-routeing problem. U. Zimmermann, U. Derigs, W. Gaul, R. H. Mohring, K. P. Schuster, eds. *Operations Research Proceedings*. Springer, Braunschweig, Germany.

Chan, Y., W. B. Carter, M. D. Burnes. 2001. A multiple-depot, multiple-vehicle, location-routing problem with stochastically processed demands. *Comput. Oper. Res.* **28** 803–826.

Chien, T. W. 1993. Heuristic procedures for practical-sized uncapacitated location-capacitated routing problems. *Decision Sci.* **24**(5) 995–1021.

Cortinhal, M. J., M. E. Captivo. 2003. Upper and lower bounds for the single source capacitated location problem. *Eur. J. Oper. Res.* **151** 333–351.

Dongarra, Jack J. 2006. Performance of various computers using standard linear equations software. Technical Report CS-89-85, University of Tennessee.

Ghiani, G., G. Laporte. 2001. Location-arc routing problems. *OPSEARCH* **38** 151–159.

Glover, F., M. Laguna. 1997. *Tabu Search*. Kluwer Academic Publishers, Boston, MA.

Labadi, N. 2003. Problèmes tactiques et stratégiques en tournées sur arcs [Tactical and strategic problems on arc routing]. Unpublished doctoral thesis, University of Technology of Troyes, France.

Laporte, G., F. Louveaux, H. Mercure. 1989. Models and exact solutions for a class of stochastic location-routing problems. *Eur. J. Oper. Res.* **39** 71–78.

Laporte, G., Y. Norbert, S. Taillefer. 1988. Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Sci.* **22**(3) 161–167.

Lin, S., B. W. Kernighan. 1973. An effective heuristic algorithm for the traveling salesman problem. *Oper. Res.* **21** 498–516.

Min, H., V. Jayaraman, R. Srivastava. 1998. Combined location-routing problems: A synthesis and future research directions. *Eur. J. Oper. Res.* **108** 1–15.

Pirkul, H. 1987. Efficient algorithm for the capacitated concentrator location problem. *Comput. Oper. Res.* **14** 197–208.

Pisinger, D. 1997. A minimal algorithm for the 0-1 knapsack problem. *Oper. Res.* **45** 758–767.

Prins, C., C. Prodhon, R. Wolfler Calvo. 2004. Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité. A. Dolgui, S. Dauzère-Pérès, eds. *MOSIM '04*, Vol. 2. Lavoisier, Ecole des Mines de Nantes, France, 1115–1122.

Prins, C., C. Prodhon, R. Wolfler Calvo. 2005. A reactive GRASP and path relinking algorithm for the capacitated location routing problem. *CD Proc. Internat. Conf. Indust. Engrg. Systems Management*, Marrakech, Morocco, May.

Prins, C., C. Prodhon, R. Wolfler Calvo. 2006. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *4OR—Quart. J. Oper. Res.* **4**(3) 221–238.

Salhi, S., G. K. Rand. 1989. The effect of ignoring routes when locating depots. *Eur. J. Oper. Res.* **39** 150–156.

Srivastava, R. 1993. Alternate solution procedures for the locating-routing problem. *OMEGA Internat. J. Management Sci.* **21**(4) 497–506.

Toth, P., D. Vigo. 2003. The granular tabu search and its application to the vehicle routing problems. *INFORMS J. Comput.* **15**(4) 333–346.

Tuzun, D., L. I. Burke. 1999. A two-phase tabu search approach to the location routing problem. *Eur. J. Oper. Res.* **116** 87–99.

Wu, T. H., C. Low, J. W. Bai. 2002. Heuristic solutions to multi-depot location-routing problems. *Comput. Oper. Res.* **29** 1393–1415.

Computers & Operations Research 38 (2011) 723–739

Contents lists available at ScienceDirect

Computers & Operations Research

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

Efficient frameworks for greedy split and new depth first search split procedures for routing problems

Christophe Duhamel [a], Philippe Lacomme [a], Caroline Prodhon [b,*]

[a] Laboratoire d'Informatique (LIMOS, UMR CNRS 6158), Campus des Cézeaux, 63177 Aubière Cedex, France
[b] Institut Charles Delaunay (UMR CNRS 6279), Université de Technologie de Troyes, BP 2060, 10010 Troyes Cedex, France

ARTICLE INFO

Available online 18 October 2010

Keywords:
Routing problem
Location routing problem
Split

ABSTRACT

Split procedures have proven their efficiency within global optimization frameworks for routing problems by splitting giant tours into trips. This is done by generating an optimal shortest path within an auxiliary graph constructed from the giant tour. This article provides a state of the art of split practice in routing problems and gives its key features. The efficiency of the method critically depends on the node-splitting procedure and on the upper and lower bound approximations. Suitable complexity can be obtained using a new depth first search procedure which is introduced here and provides a new algorithm that is specially designed for large scale problems with resource constraints. Experiments show that the depth first search split procedure introduced in this article, used as an evaluation function in a global framework, can improve the results obtained by the use of the "classical" split procedure on the Location-Routing Problem and the Heterogeneous Vehicle Routing Problem. The Location-Routing Problem is also used to provide further analysis and a fair comparative study between the two split versions.

1. Introduction

1.1. Routing problems

The generic problems under consideration in this paper are related to the class of routing problems. In routing problems, the aim is generally to service a set of customers thanks to trips performed by vehicles that travel through routes visiting the customers at least cost.

Formally it can be described in terms of the following components: the network, the demand(s), the fleet, the cost(s), and the objective(s).

The *network* can be symmetrical, asymmetrical or mixed. It is represented as a graph on which the nodes model towns, customers and/or depots; and arcs model real or symbolic connections (e.g. roads, pipelines). Most routing problems consider only one depot node.

Demands generally represent the delivery or collection of products, associated with tasks on nodes and/or arcs. If the nodes must be serviced, the routing problem is referred to as a Node Routing Problem and if the arcs must be serviced the problem is referred to as an Arc Routing Problem. Demands on tasks can be fixed or stochastic and can be defined in terms of different product types. Most routing problems consider that products are gathered at the end of the trip which is a special case of the wide-ranging pick-up and delivery routing problems (PDVRP). In PDVRP each demand on tasks has a specific destination task.

The vehicle *fleet* can be either heterogeneous or homogeneous. In the context of the VRP, vehicle fleet characteristics have led to several well-known routing problems including the vehicle fleet mix problem (VFMP) and the heterogeneous VRP (HVRP). In VFMP, the fleet is composed of t vehicle types. Each type k is defined by a capacity Q_k, a fixed cost f_k and a cost per distance unit v_k, often called variable cost. A trip of length L made by a vehicle of type k has a cost $f_k + L \times v_k$. The number of vehicles of each type is not limited. The HVRP is a further generalization with a limited availability a_k for each vehicle type k.

Costs are generally fixed for the set-up of depots and of vehicles, and variable in terms of distance travelled or time used. They can include service penalties incurred when a customer receives a late or incomplete delivery. Related to costs, profit can also be associated to given tasks when servicing them (orienteering problem).

Objectives can be multiple and diverse. The objective function can be computed for a single or for several periods (periodic VRP—PVRP), though in the latter case, both vehicles and visits must be assigned to the different periods. The most common objective aims at minimizing a solution cost. However, it can include the minimization of the total travelled distance, the total required time, the total tour cost, and/or the fleet size, and the maximization of the service quality and/or the profit collected. When multiple objectives are identified, they are frequently in

* Corresponding author.
E-mail address: caroline.prodhon@utt.fr (C. Prodhon).

0305-0548/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2010.09.010

conflict. For this reason adopting a multi-objective point of view can be advantageous.

1.2. Split methods

The *split* method was originally proposed by Beasley [1] in 1983 as the second phase of a route-first, cluster-second heuristic to solve the Vehicle Routing Problem. The first phase computes a giant tour of all customers (solving a Travelling Salesman Problem—TSP), by relaxing vehicle capacity and maximum tour length. The second phase constructs a cost network and then applies a shortest path algorithm to find least cost feasible trips. Apart from a theoretical interest for proving some worst-case performance ratios, this first step in split approaches is limited to heuristics. A similar approach based on breaking the travelling salesman tour into disjoint segments is proposed in 1985 by Haimovich and Rinnooy Kan [2]. It is applied by Golden et al. [3] for the Heterogeneous Fleet VRP (HVRP). Ulusoy [4] implements this heuristic for the fleet size and mix problem for the capacitated arc routing problem, Ryan et al. [5] and Renaud et al. [6] for the VRP in 1993 and 1996, and Salhi and Sari [7] in 1997 for the multi-depot HVRP. The second step in split approaches aroses in 2001 when the split procedure was implemented within a general framework for routing problems through a memetic algorithm for the Capacitated Arc Routing Problem (CARP). This memetic algorithm introduced by Lacomme et al. [8,9] competes with the best published methods from 2001 to 2008 on the CARP providing better results than the very efficient CARPET method [10] on the three sets of instances used for the CARP (DeArmon's instances, Belenguer's instances and Eglese's instances). In 2004, Prins introduced an efficient method (with regard to the two sets of instances used, we mean Christofides' instances and Golden's instances) for the VRP also taking advantages of the split procedure [11]. In this context, the number of split applications in routing increases, using it either on its classical version or on a specific one, and leads to solving further problem extensions as shown in subsequent papers.

The Periodic Arc Routing Problem has been efficiently solved through an evolutionary algorithm [12] and a scatter search [13]. Generally the objective functions are designed to optimize one single criterion including, for instance, the transportation cost. However, split based frameworks have a wide range of applications including, but not limited to, bi-objective and stochastic optimization problems. In 2006, Lacomme et al. [14] proposed the first resolution of the bi-objective CARP and in 2005 the Stochastic CARP was solved by Fleury et al. [15]. In addition, the Stochastic CARP resolution was realized using a bi-objective scheme in 2008 [16]. Time window constraints are included in the CARP that is efficiently solved using a GRASP/Path-Relinking framework [17]. Finally, the split-delivery capacitated Arc Routing Problem was addressed by Labadi et al. in 2008 [18] and the Cumulative CARP (the objective being to minimize the sum of arrival times at customers, instead of the classical route length) by Ngueveu et al. [19]. One can note the recent publication of Santos et al. [20] which presents an improved heuristic for the CARP also based on a split procedure. A full bibliography on CARP is proposed by Corberan and Prins [21].

The basic VRP has been efficiently solved using split based framework by Villegas et al. in 2008 [22]. In 2009, Prins et al. [23] proposed some algorithms for the VRP still based on split procedures. Multi-Compartments routing problems (MC-VRP) (in which products can be stored in several dedicated sections within a same vehicle) were addressed in 2008 by El Fallahi et al. [24] with a memetic algorithm including a post-optimization phase based on path relinking, and a tabu search method. Numerical experiments encompass small and large scale instances from 50 to 484 customers and two compartments with an average computational time around 3 min for the tabu search and 2 min for the memetic algorithm. Uncertainty on the demand for the MC-VRP seems to have been largely ignored in academic literature until 2010. We have only come across the recent publication of Mendoza et al. [25] who extended the split procedure to this case with a stochastic MC-VRP. VRP incorporating Time-Windows was considered by Labadi et al. in 2008 [26]. Imran et al. [27] solve the HVRP thanks to a variable neighborhood-based heuristic. The publication of Prins [28] in 2009 also deals with heterogeneous fleets of vehicles and uses the last extension of the split procedure taking into account resource constraints relative to the fleet. Numerical experiments carried out on the whole VFMP and HVRP instances prove that the method outperforms all previously published methods on HVRP using the classical well-known benchmark instances. Concerning the simultaneous nodes and edges routing problems, the publication of Prins and Bouchenoua [29] is the first integrating a split with this extension. Split approaches are also part of VRP extensions such as the split delivery [30], the pick-up and delivery [31,32], the two-dimensional loading [33], the team orienteering problem [34] and customer demands including, but not limited to, Dial-a-Ride [35].

The Location Routing Problem (LRP) was first addressed with a classical split procedure in 1996 [36]. Nagy and Salhi [37] introduced the many-to-many LRP, in which each customer location is considered as a feasible depot site, and two classes of vehicle are available: inter-hub and local delivery/collection vehicles. In 2006, the split principle was embedded within a memetic algorithm by Prins et al. [38]. In 2008, Duhamel et al. [39] also introduced a memetic algorithm using a tuned version of the procedure which encompasses specific extra constraints including limitation on hub capacity and a heterogeneous set of hubs. Lately Duhamel et al. [40] in 2009, presented a very efficient GRASP-ELS approach for the LRP. Extensions of LRP are also tackled by split based approaches. These extensions include periodic demands in [41] and the Truck and Trailer Routing Problem [42,43], which can be seen as a two-level LRP without capacities on the depots and on the vehicle supplying the depots from the main factory.

Table 1 sums up the use of split procedure in routing problems.

In this paper, we present all the key features in a split procedure to address routing problems with extra-constraints. Our presentation spans the whole process including the challenge associated with the definition of a very efficient split algorithm framework, and investigates a new depth first search split procedure. We illustrate these features using both the Location Routing Problem (LRP) and the Heterogeneous Vehicle Routing Problem (HVRP) with numerous numerical experiments.

2. Split procedure in routing problems

2.1. Efficient routing framework principles

Routing problems are widespread in distribution and logistics. In a majority of cases, they consist of determining a set of least-cost trips to service tasks on a weighted graph under a set of constraints such as vehicle capacity or the number of vehicles available for each type.

Efficient frameworks include metaheuristics such as memetic algorithms, GRASP or tabu search. In addition, concerning routing problems previous studies (see Section 1.2) push into accepting that the search space should be managed through compact representations favouring a partial enumeration of the whole search space.

Table 1
Main published frameworks for routing problems using a split procedure.

References	Routing problem	Comments
CARP and extensions		
Lacomme et al. [8]	CARP	First application of split in a framework
Lacomme et al. [9]	CARP	Memetic algorithm
Santos et al. [20]	CARP	Improved heuristic
Ulusoy [4]	Fleet size and mix problem for the CARP	Heuristic
Lacomme et al. [12]	Periodic CARP	Evolutionary algorithm
Chu et al. [13]	Periodic CARP	Scatter sarch
Fleury et al. [15]	Stochastic CARP	Genetic algorithm
Lacomme et al. [14]	Bi-objective CARP	Genetic algorithm
Fleury et al. [16]	Bi-objective Stochastic CARP	NSGA II
Labadi et al. [17]	CARP with time-windows	GRASP with path relinking
Labadi et al. [18]	Split delivery CARP	Evolutionary algorithm
Ngueveu et al. [19]	Cumulative CARP	Memetic algorithm
VRP and extensions		
Beasley [1]	VRP	First historic split procedure
Haimovich and Rinnooy Kan [2]	VRP	Heuristic
Ryan et al. [5]	VRP	Heuristic
Renaud et al. [6]	VRP	Heuristic
Prins [11]	VRP	Genetic algorithm
Villegas et al. [22]	VRP	Improved framework
Prins et al. [23]	VRP	Memetic algorithm and Tabu search
El Fallahi et al. [24]	MC-VRP	Efficient resolution of a hard to solve extension of the VRP
Mendoza et al. [25]	Stochastic MC-VRP	Genetic algorithm
Labadi et al. [26]	VRP with Time-Windows	Memetic Algorithm
Golden et al. [3]	HFVRP	Heuristic
Salhi and Sari [7]	HFVRP	Heuristic
Imran et al. [27]	HVRP	Variable neighbourhood heuristic
Prins [28]	HVRP	Memetic algorithms
Prins and Bouchenoua [29]	Split Delivery VRP	Memetic Algorithm
Boudia et al. [30]	Split Delivery VRP	Memetic Algorithm with Population Management (MAPM)
Mosheiov [31]	Pick-up and Delivery problem	Heuristic
Velasco et al. [32]	Pick-up and Delivery problem	Memetic Algorithm
Duhamel et al. [33]	2L-CVRP	Evolutionary Local Search (ELS)
Bouly et al. [34]	Dial-a-Ride	Memetic Algorithm
Lacomme et al. [35]	Dial-a-Ride	Memetic Algorithm with Population Management (MAPM)
LRP and extensions		
Nagy and Salhi [36]	LRP	First application to LRP—Heuristic
Prins et al. [38]	LRP	Memetic Algorithm with Population Management (MAPM)
Duhamel et al. [39]	LRP	Memetic Algorithm
Duhamel et al. [40]	LRP	GRASP × ELS
Nagy and Salhi [37]	Many-to-many LRP	Heuristic
Prodhon and Prins [41]	Periodic LRP	Memetic algorithm
Villegas et al. [42]	Single Truck and Trailer Routing Problem	GRASP × VND, ELS
Villegas et al. [43]	Truck and Trailer Routing Problem	GRASP × ELS

Considering these observations, an efficient routing metaheuristic based framework should involve (see Fig. 1):

- A *quasi-direct representation of solution* (QDRS) that is not a whole solution of the problem but a compact representation, such as a sequence of nodes or tasks (called a chromosome if an evolutionary algorithm is used).
- A *function f* which associates a QDRS to a whole solution with the evaluation of one or several objective functions (depending on the number of criteria to evaluate). In this context, $\overline{f}$ denotes the function associating a solution to a quasi-direct representation (QDRS).
- A *local search* (LS) performing local improvements of a solution using, for instance, well-known basic moves such as exchanges, 2-OPT or *k*-OPT.
- A *diversification process* to generate periodically new diverse solutions under the QDRS, avoiding a premature convergence to local minima.
- *Heuristics* to quickly generate good initial solutions, e.g. Clarke and Wright's heuristic and Golden et al.'s heuristic that are widely used in VRP.

The key-point for the efficiency of such metaheuristics is to alternate between two solution spaces. The QDRS space exploration is devoted to metaheuristics trying to avoid local minima and clone generation, while the routing solution space is used to intensify the search by improvement through a local search. There, a clone generation refers to the generation of a new and different compact solution representation in the QDRS space, which leads to an already visited routing solution.

In routing problems a QDRS can be a giant tour λ visiting all servicing tasks (nodes or arcs), i.e. a permutation of n tasks (TSP tours on the n customers). Then, the routing solution can be retrieved by splitting this sequence into trips through a resource-constrained shortest path problem or a basic shortest path (depending on the routing problem characteristics) on an auxiliary graph.

The lower part of Fig. 1 depicts the successive changes of encoding induced. With the split approach, a giant tour λ (QDRS) can be converted by *Split* (f) into a routing solution S, with respect to the given sequence. A local search can be efficiently defined for S providing an improved solution S'. Finally, a *Concat* procedure ($\overline{f}$) can convert S' into a giant tour λ' (QDRS), by concatenating its trips. Even without extra operators from the framework (mutations, crossover, etc.), several successive improved solutions

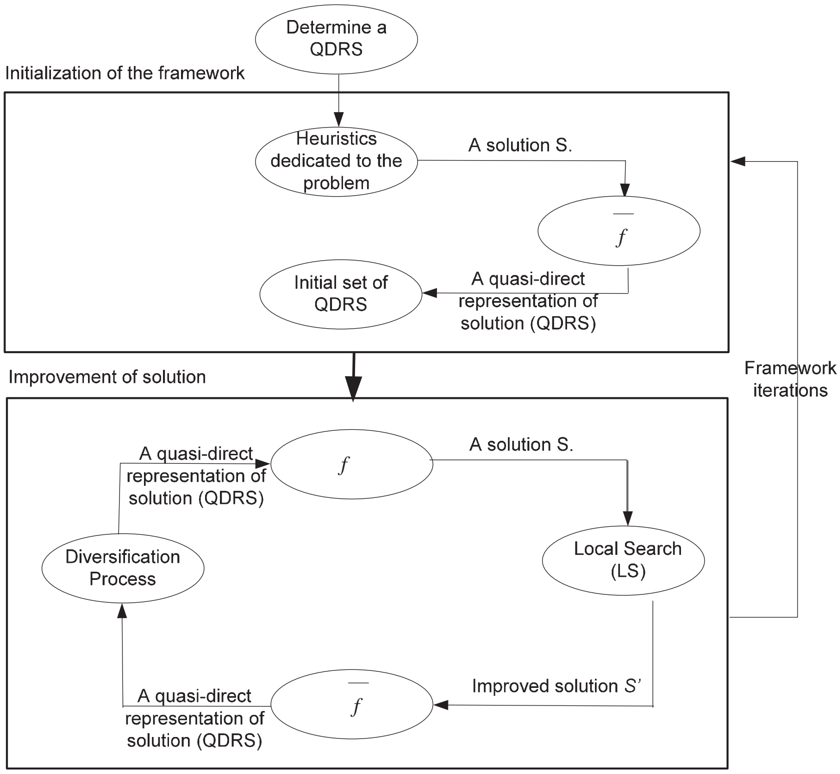

Fig. 1. Efficient routing framework outlines.

can be obtained, because $\lambda' \neq \lambda$ in general. All previous routing frameworks based on a split approach address the previously introduced key points. However, the distinction between these points and the guideline is not apparent in all publications since authors focus on their contribution (constructive heuristic or new metaheuristic for search space exploration).

2.2. Split outlines

Before explaining how the split procedure works, one needs to define some notations.

2.2.1. Notations and definition

Let us consider a routing problem to be solved with $F=1,\ldots f$ criteria to minimize, subject to $T=1,\ldots,t$ types of resource constraints (e.g. depots for the LRP or vehicles for a HVRP). Each type of resources p has from 1 to r_p different sub-types. For a problem with a heterogeneous fleet of trucks, a type of resource can be the vehicles, and the sub-types would be the different available trucks/capacities.

In the split-based framework, $\lambda=(1,\ldots,n)$ is used as a quasi-direct representation of solution (giant tour), where $\lambda=(i)$ is a serviced task.

The *split procedure* consists of building an auxiliary acyclic graph H with $n+1$ nodes numbered from 0 to n where an arc from node i to j represents a subsequence μ_{ij} of λ with $\mu_{ij}=(\lambda(i+1),\ldots,\lambda(j))$.

A subsequence μ_{ij} represents a "feasible trip", i.e. a trip for which all the problem constraints are satisfied. The trip consists of routing from a depot to task $\lambda=(i+1)$, from $\lambda=(i+1)$ to $\lambda=(i+2)$ and so on until $\lambda=(j)$ before coming back to the departure depot from task $\lambda=(j)$. Depending on the problem to be solved, it is possible to assume that the least-cost paths between tasks have been pre-computed taking into account extra constraints including time-windows on tasks for example. The optimal splitting of λ (or quasi-optimal splitting in case of extra constraints concerning the use of resources) corresponds to a min-cost path from node 0 to node n in H. It can be computed in polynomial time using Bellman's algorithm for directed acyclic graphs (classical version). If resource constraints must hold (extended version), several labels have to be tackled per node [44].

Therefore, to extract the trips from the giant tour, a set of non-dominated labels is saved on each node j of the graph H. A label represents a partial solution evaluation tackling tasks $\lambda=(l)$–(j). Let us note L_i^j the ith *label* on node j. It is composed of a p-uplet P_i^j, a couple (u,v) and a status S_i^j:

- A p-uplet P_i^j defining both cost and resource availability $P_i^j = (C^1_{L_i^j}, \ldots, C^f_{L_i^j}, Q_1^1(L_i^j), \ldots, Q_{r1}^1(L_i^j), \ldots, Q_1^t(L_i^j), \ldots, Q_{rt}^t(L_i^j))$ where $C^l_{L_i^j}$ is the current value at node j of the lth *criteria* of the objective function for the label L_i^j, and $Q_q^p(L_i^j)$ is the availability of the qth resource of type p for the label L_i^j.
- a *couple* $(u,\ v)$ of position referring to the father label of L_i^j, i.e. L_i^j which has been generated using the uth label of node v.
- a *status* $S_i^j \in (true,\ false)$ indicating if the propagation of the label has been achieved or not.
- Depending on L_i^j and on μ_{jm}, $A^k_{\mu_{jm},L_i^j}$ denotes the kth *feasible assignment* of the trip μ_{jm} according to the resources available at the label L_i^j. Note that $A_{\mu_{jm},L_i^j} = \{A^k_{\mu_{jm},L_i^j}\}$ is the set of all possible assignments of a trip μ_{jm} using label L_i^j.

$T^k_{\mu,L}$ is the p-uplet linked to $A^k_{\mu,L}$ the kth feasible assignment of the trip μ using the father label L. $T^k_{\mu,L}$ is a *trip evaluation* fully defined by a p-uplet defining both the trip cost and the resource consumption: $T^k_{\mu,L} = (V^1_T, \ldots, V^f_T, R^1_1(T), \ldots, R^1_{r1}(T), \ldots, R^t_1(T), \ldots, R^t_{rt}(T))$ Formally, $V^l_{T^k_{\mu_{jm},L^j_i}}$ is the cost of the subsequence μ_{jm} as regards the lth criteria. R^p_q is the consumption of resource $q \in 1 \ldots r_p$ of type $p \in 1 \ldots t$.

A *propagation function* defined by $f: L \otimes T \rightarrow L$ permits the generation of a new label L^m_k from one label L^j_i and a trip evaluation $T^k_{\mu_{jm},L^j_i}$. Typically costs are updated by basic addition and resource constraints by diminution of the resource availabilities.

$$\forall s = 1, \ldots, f, \;\; C^s_{L^m_k} = C^s_{L^j_i} + V^s$$

$$\forall p = 1, \ldots, t, \;\; \forall q = 1, \ldots, r_p, \quad Q^p_q(L^m_k) = Q^p_q(L^j_i) - R^p_q(T^k_{\mu_{jm},L^j_i})$$

If $\forall p = 1, \ldots, t, \; \forall q = 1, \ldots, r_p, \quad Q^p_q(L) > 0$, the initial label L is responsible for the generation of $\prod_{p=1}^{t} r_p$ new final labels on the destination node.

These final labels will be included or not depending on the set of non-dominated labels previously saved on the destination node and their potential.

To keep only non-dominated labels on nodes, a *dominance rule* must be defined. If a single criterion is used on trips and nodes, if this criterion is an integer-valued function, and if no resource availability is addressed, the classical "strictly smaller" operator permits the comparison of two labels. A similar remark holds for a real-valued function, except that from a computer engineering point of view, the comparison must be $|a-b| < \varepsilon$ where ε is a tiny positive number. For the general label definition, the dominance rule does not define a complete order of labels and a dominance function is required.

Let us consider two labels L and P. L is defined as dominant as regards P ($P \prec L$) if one the following conditions holds:

Condition 1.
$\exists s \in [1, \ldots, f] / C^s_L < C^s_P$ and $\forall j \neq s / C^j_L \leq C^j_P$ and $\forall p \in [1, \ldots, t], \forall q \in [1, \ldots, r_p] / Q^p_q(L) \geq Q^p_q(P)$.

Condition 2.
$\exists p \in [1, \ldots, t], q \in [1, \ldots, r_p] / Q^p_q(L) > Q^p_q(P)$ and $\forall j \neq p, \forall k \neq q / Q^j_k(L) \geq Q^j_k(P)$ and $\forall s = 1, \ldots, f / C^s_L \leq C^s_P$

If L is not dominant as regards P ($P \overline{\prec} L$), this does not imply that P is dominant as regards L. If $P \overline{\prec} L$ and $L \overline{\prec} P$, then L cannot be compared to P.

Thus, if $\exists k / L \prec L^k_i$, then L is not added to node i. On the other hand, each label $L^k_i / L \succ L^k_i$ is removed from node i. An *efficient sub-routine* denoted `dominate` (see Algorithm 1) can be implemented as follows: given two labels L and P on a node, it is possible to define the function `dominate` providing result 1 if $L \succ P$, 2 if $L \succ P$ and 3 otherwise. If only one criteria is linked to a label ($C^1_{L^j_i}$), saving labels in increasing cost order using ($C^1_{L^j_i}$) allows a label. insertion and removal in $O(N_L/2)$.

Furthermore, in addition to the dominance rule on node i, for each label L generated on this node, it is possible to evaluate its potential by investigating whether the insertion is *promising* or not, i.e. if the label L could induce a non dominated solution on the final node in the graph. This could be done through a procedure, called for instance `check_if_promising`, in which a lower bound LB_i evaluating the service of the nodes located after i in λ should be computed. It must at least tackle the different criteria and may also include the resource consumption. If information from both the considered label L on node i and LB_i appears leading to a worst situation than an upper bound on node n (solution found by a heuristic or a previously computed label on node n), L is said to be not promising and should be discarded.

In the overall split algorithms (Algorithms 2 and 3), we integrate both sub-routing, `dominate` and `check_if_promising`, within a procedure, that we call `Insertion`, to test the interest of a label.

2.2.2. Example of dominance for the HLRP

Let us consider a heterogeneous location routing problem (HLRP) in which two objective functions have to be minimized: the total routing cost as well as the maximal duration of a route. Thus, $|F| = 2$. Assume that 2 classes of vehicles are available. The first one is composed of 4 vehicles with a capacity of 10 t. The second one has 3 vehicles of 15 t. The routing must be achieved with a subset of 3 depots with a capacity of 40 t for depot 1, 45 t for depot 2 and 50 t for depot 3. Thus the resource constraints are the vehicle fleet and the depots leading to $|T| = 2$. Concerning the first set of constraints (vehicles), $r1 = 2$ vehicle sub-types and their limits correspond to the number of vehicles
of each size (4 and 3). For the second set of constraints (depots), one resource sub-type can be one depot ($r2 = 3$, the number of available depots) and its limit is its capacity (40, 45 and 50).

Label $L^j_i = (1000, 15; 3, 1; 30, 45, 30; 2, 4; \mathit{False})$ means that to serve nodes $\lambda = (l)$–$\lambda = (j)$ from this label, the travel cost is 1000 and the maximal duration of a route is 15. Furthermore, 3 vehicles of capacity 10 t and 1 having a capacity of 15 t remain (thus, $4-3=1$ vehicle of type 1 and $3-1=2$ vehicles of type 2 have been used). The capacities still available on the depots are 30, 45 and 30, meaning that depot 2 has not yet been opened (the availability is equal to the capacity, i.e. 45). The father of L^j_i is the 2nd label on node 4. Finally, *False* means that the propagation of the label L^j_i has not been achieved yet. L^j_i is said to be "dominant" as regards $L^j_k = (1000, 15; 3, 0; 30, 45, 30; 1, 5; \mathit{False})$ since for the same cost and maximal route duration, L^j_i has one more vehicle of type 2 available.

Algorithm 1. Dominance rule.

```
function dominate (L,P): integer;
//assume L and P are not comparable
LDomP: = true
PDomL: = true
for i: = 1 to f do
  if(L.C(i) > P.C(i)) then LDomP: = false endif
  if(P.C(i) > L.C(i)) then PDomL: = false endif
endfor
j: = 1
while (j < = t) and ((LDomP = true) or
        (PDomL = true)) do
    k: = 1
    while(k < = r(j)) and( (PDomL = true)
    or(LDomP = true)) do
      if(L.Q(k) < P.Q(k)) then LDomP: = false;
       endif
      if(P.Q(k) < L.Q(k)) then PDomL: = false;
       endif
      k: = k+1;
    endwhile
  j: = j+1;
endwhile
if ((LDomP = true) and (PDomL = false)) then
return 1
    elseif((PDomL = true)and(LDomP = false))
    then return 2
      elseif((LDomP = true)and(PDomL = true))
      then return 2
        else return 3
```

2.2.3. Split parameters

As stressed above, when dealing with resource consumptions, the computation of a min-cost path from node 0 to node n in H

requires the management of several labels per node. Avoiding a huge number of labels per node should speed up the algorithm. Thus, in addition to a definition of an upper bound on the final node and a lower bound on each node *i* evaluating the service of the nodes located after *i* in λ, it could be useful to introduce the following parameters for a split procedure:

N_L: the maximal number of labels saved on each node;
N_{max}: the maximal number of labels generated during the split algorithm.

The ratio from the split efficiency as regards the computational time is highly dependent on the N_L and N_{max} parameters. A tiny or moderate value of N_{max} can lead to a premature stop of the method inducing, in general, a weak evaluation for the DFS split and a greedy split fail (see Sections 2.3 and 2.4). Similar remarks hold for N_L, for which a tiny value can induce a weak evaluation of the sequence λ. On the other hand, increasing the number of labels per node does not guarantee an improvement in the solution quality. Indeed, for a solution cost obtained with a given value of N_L, the trips are obtained by a careful scan of the father labels from the end of the graph to the beginning. Increasing the number of labels per node modifies the list of labels saved at each step of the split. For one node of the graph, a label *L* which has been saved for a smallest value of N_L, can be now discarded amending the subsequent steps of the split. The consequence can be the non-exploration of the shortest path previously obtained when the number of labels per node was inferior. This could lead sometimes to a worst split solution.

Note that, from a theoretical point of view, the split procedure is optimal only if the total number of labels and the maximal number of labels per nodes are not upper bounded which is unpractical for routing problems with extra constraints including but not limited to LRP, HVRP, etc.

So, the difficulty consists in tuning the split parameters to obtain a time efficient split procedure with first-rate trips cost. Any fair comparative evaluation must consider:

- the quality depending on (N_{max}, N_L) for a fixed sequence $\lambda=(1,\ldots,n)$ of serviced nodes;
- the convergence rate of the framework which strongly depends on the split procedure.

Algorithm 2. Generic Greedy split algorithm.

procedure Greedy_Split(λ=(1,…,n));
Build the graph G.
L=(0,…,0,$R_1^1(t_{\mu,L}^k)$,…,$R_{r1}^1(t_{\mu,L}^k)$,…,$R_1^t(t_{\mu,L}^k)$,…, $R_{rt}^t(t_{\mu,L}^k)$,0, 0, false
Add_Label_To_Node (G,0,L)
//an upper bound is computed of the f criteria to minize
UB:=Call Compute_Upper_bound(G);
//the label UB is added to the last graph node
call Add_Label_To_Node (G, n, UB);
//for each node of the graph, a lower bound is computed
for each i=1..n, LB(i)=Compute_Lower_Bound(G)
//total number of label
Total_label:=0;
i=1
while(i<n) do
  repeat
  j:=i
  if(j=i) then Initialize (C^1,…,C^f); μ={λ(j)};
  else update (C^1,…,C^f); μ=μ+{λ(j)};
  endif
  for l:=1 tolast_label onnode I do
  Current_Label=L_i^l
    for all feasible assignment in $A_{\mu,L_i^l}=A_{\mu,L_i^l}^k$} do
      Compute T_{μ,L_i^l}
      final_Label=f(L_i^l,T_{μ,L_i^l})
      //one more label generated
      Total_label:=Total_label+1;
      if(Total_Label=N_{max}) then
      break; //premature split stop
      endif
      res:=Call Insertion (G, j, final_Label, LB(i));
      if (res=true) then
      callAdd_Label_To_Node (G, j, final_label);
      endif
    endfor
  endfor
  j:=j+1
  until(j>n) or ((C^1,…,C^f) does not comply with resource availability);
i:=i+1;
endwhile

Algorithm 3. Generic DFS split algorithm.

procedureDepth_First_Search_Split (λ=(1,…,n));
Build the graph G.
L=(0,…,0,$R_1^1(t_{\mu,L}^k)$,…,$R_{r1}^1(t_{\mu,L}^k)$,…,$R_1^t(t_{\mu,L}^k)$,…, $R_{rt}^t(t_{\mu,L}^k)$,0, 0, false
Add_Label_To_Node (G,0,L)
//an upper bound is computed of the f criteria to minimize
UB:=Call Compute_Upper_bound(G);
//the label UB is added to the last graph node
call Add_Label_To_Node (G, n, UB);
//at each node of G, a lower bound is computed
Compute_Lower_Bound(G);
//total number of label
Total_label:=0;
//initialize stack
call initialize(P);
//push the initial reference
call push(P, 0, 1) //first label on node 0
//first label to generate
start_label:=1; i=1
while(check_if_empty(P)=false) do
  repeat
  j:=i
  if(j=i) then Initialize (C^1,…,C^f); μ={λ(j)};
  else update (C^1,…,C^f); μ=μ+{λ(j)};endif
  res:=false;
  for l:=start_label to last_label on node i do
    Current_Label=L_i^l
  for all feasible assignments in $A_{\mu,L_i^l}=A_{\mu,L_i^l}^k$} do
      Compute T_{μ,L_i^l}
      final_Label=f(L_i^l,T_{μ,L_i^l})

```
30.       Total_label:=Total_label+1;
31.       if(Total_Label=N_max) then break; endif
32.       res:=Call Insertion (G, j, final_Label,
          LB(i));
33.       if(res=true) then
34.         call Add_Label_To_Node (G, j,
            final_label);
35.         if(j=n) then Save_G:=G; endif;
36.       endif
37.     endfor
38.   endfor
39. if (res=true) then
40.     for k:=1 to last_label onnode jdo
41.       if (L_j^k·s=true) then call push(P, j, k);
          endif;
42.     endfor
43.   endif
44. j:=j+1
45. until(j>n) or ((C^1,...,C^f) does not comply with
    resource availability);
46. Stop:=false;
47. While(check_if_empty(P)=false) and
    (stop=false) do
48.     pop(P, i, start_label);
49.     if(i<n) then stop:=true; endif;
50. endwhile
51.endwhile
```

2.3. Greedy split outlines

In the classical greedy split procedure, thanks to two indices i and j, the Algorithm 2 enumerates each subsequence μ_{ij} of $\lambda=(1,\ldots,n)$ that corresponds to a feasible trip and computes its costs and resource consumptions. Firstly, an initial label L is added to node 0 with all objective functions stated at 0 and all resource availabilities set at their maximal value.

Secondly, both an Upper Bound UB on the last node and a Lower Bound LB_i on each node i is computed.

The `while` loop (lines 14–39) permits the enumeration of all starting nodes for a trip whereas `repeat...until` loop (lines 15–37) enumerates all ending nodes for an initial node i. The `repeat ...until` loop stops as soon as the ending node is reached or when the current trip resource consumptions exceed the availabilities for one or more resources.

The third loop (lines 20–35) enumerates all labels on node i.

The trip μ_{ij} evaluation is investigated over all possible assignments on resources and each satisfactory assignment leads to an evaluation and finally to the generation of the `final_Label`. If the `final_label` is stated by the procedure `Insertion` as promising and not dominated, it is added to node j thanks to a call to `Add_Label_To_Node`.

2.4. Depth first search split outlines

The depth first search split (DFS split) procedure (Algorithm 3) is designed initially to investigate the labels close to the final node trying to reach this ending node as quickly as possible.

While the greedy split algorithm is managed using a loop from 1 to n, the DFS algorithm iterates until a stack of labels is empty or a maximal number of labels to be generated is reached. The initial stack configuration is composed of the couple (0,1) representing label 1 on node 0. The trip μ_{ij} leads to new labels on node j and when all the labels on node i have been propagated, the new labels are saved on stack using the push procedure. The index i is updated using the couple at the top of the stack. The `pop` procedure is iterated until a couple referring to a node from 1 to n is obtained thanks to lines 47–50 of the algorithm.

Note that each time a new best label is saved on node n, the whole split graph (with the current labels) is saved using `Save_G:=G`. This is necessary since the labels on the incumbent graph would be updated during the algorithm backtracks and the shortest path from node 0 to n would be lost. Such a state cannot occur for the greedy split algorithm for which no backtrack is performed. The shortest path must be retrieved using `G:=Save_G` at the end of the algorithm.

3. A GRASP × ELS based on split methods for the LRP

Trying to investigate the behaviour of both split procedures, a GRASP × ELS framework is introduced for the Location-Routing Problem taking advantage of the split procedures and the key features we highlight in Section 2.1.

3.1. The location routing problem

The LRP aims at simultaneously determining the location and routing decisions. LRP can be classified (see [45]) taking into account depot capacities, vehicle fleet characteristics (homogeneous or heterogeneous) and the costs of vehicles. It is possible to refer to the last published survey of Nagy and Salhi in 2007 [46]. The LRP with homogeneous or with limited heterogeneous fleets was studied by Wu et al. [47]. Prins et al. [48] considered the Capacitated LRP with a homogeneous and unlimited fleet of capacitated vehicles and capacitated depots. They proposed a Greedy Randomized Adaptive Search Procedure (GRASP) complemented by a Learning Process on the depots to use and a Path Relinking. The current best published method (LRGTS) for the Capacitated LRP was introduced by Prins et al. in 2007 [49]. In 2010, Duhamel et al. [40] presented a *GRASP × ELS* framework which is competitive with LRGTS and provides new best solutions. The latter approach uses a greedy split procedure dealing with resource constraints to determine the assignment of the trips to depots. Trying to promote a fair comparative study, exactly the same framework is used here with both the greedy split and the depth first search split. Using precisely the same GRASP × ELS framework, any numerical variation in the results can only be due to the split procedure which is the only part of the code updated.

3.2. GRASP × ELS framework

GRASP × ELS [50] is a hybridization of a Greedy Randomized Adaptive Search Procedure (GRASP) with an Evolutionary Local Search (ELS). It takes advantage of both methods:

- the multi-start approach of the GRASP [51,52], based on a greedy randomized heuristic that provides np initial solutions, improved by a local search;
- the efficiency of ELS [53], an extension of the Iterated Local Search (ILS). A classical ILS builds ni successive solutions, creating at each iteration one child-solution, using mutation and local search processes. ELS is similar but generates $nc>1$ child-solutions at each iteration and selects the best one. The purpose of ELS is to sample the attraction basins close to the current local optimum better, before leaving it.

A full description of the GRASP × ELS framework is introduced in [50] which includes a description of the algorithms in use.

The key point consists of swaps between solutions encoded as giant tours and complete CLRP solutions. The split procedure is a part of the `Evaluate` procedure which links a giant tour to a CLRP solution.

3.3. Label definition for split procedures

In both the greedy and DFS split, a label L_i^j is fully defined by $(C, Q_1, \ldots, Q_{nd}, u, v, S_{L_i^j})$, where c is the label cost, and Q_j the remaining capacity for hub j, with nd the number of hubs. For example $L_i^j = (500, 10, 5, 10, \ldots)$ means that customers $\lambda = (l), \ldots, \lambda = (j)$can be serviced at a cost of 500 and a capacity of 10 remains for hub 1, 5 for hub 2 and 10 for hub 3.

The current label L_i^l is used to generate the `final_label` which is inserted only if it is promising. Two conditions have to be checked:

- its cost added to a lower bound must be lower than the cost of the upper bound on the last node;
- its maximal residual capacity over the m hubs must be enough to serve at least the demand of the biggest remaining customer. Note this second condition is strongly efficient as stressed in [40].

The main interest here is that this label definition allows the split procedures to manage the depot location directly through the assignment of the trips, as it is well-known in the LRP literature that opening or closing depots are the most critical decisions. It is theoretically possible to design a local search combining classical VRP moves (such as customer exchanges) and depot opening-closing moves, but the latter are very tricky to design. For instance, before closing a depot, its customers must be reallocated to other depots, leading to ejection chains which can fail and require time-consuming backtracking. This is why most published meta-heuristics separate the two kinds of moves in some way. In [49] for instance, a facility-location phase computes a set of open depots, so that the routing phase reduces to a multi-depot VRP.

Furthermore, intensification is achieved within the GRASP $\times$ ELS framework proposed in [40] by reducing the set of available depots during the computation of the shortest path in the auxiliary graph (extra details are proposed in [40]).

3.4. Benchmarks of interest

Prodhon's instances [54] (available at http://prodhonc.free.fr/) are the only instances where there is a significant gap between the best known solutions and the lower bounds and this is the reason why we focus on these instances.

In addition, we introduce a new set of instances, the objective being to benchmark our framework in more pragmatic conditions. In fact, location routing instances used for years in research are large scale ones but they are based on Euclidean distances which does not correspond to realistic conditions. The new instances can be downloaded with a user friendly graph drawing tool at: http://www.isima.fr/~lacomme/lrp/lrp.html and http://prodhonc.free.fr/

This set of instances is based on asymmetric sparse graphs plus a strongly heterogeneous set of depots and non Euclidean distances from nodes. Depending on the instances, the number of hubs varies from 3 for the smallest instances to 23 for the largest ones. A solution is defined by an ordered set of arcs and nodes including both servicing and non servicing nodes. Identical sub-sequences of nodes can appear in several trips since two shortest paths in a non-complete graph can have a list of nodes in common. Note also that a servicing node can be a part of a trip where it is not serviced but where it is part of a shortest path between two servicing nodes in the trip.

To understand better the specificities of the new instances, we can look more closely at a particular one: DLPP90. It is composed of 90 nodes, 292 arcs and 35 service nodes using up to 5 hubs. The hub opening costs are extremely heterogeneous, varying from 385 for hub 1 to 2301 for hub 3, and are not related to hub capacity (see Appendix C). This is a well-known situation where the hub opening cost depends on both hub capacity and position. Hub positions in city centres are presently more costly than in suburbs. The truck capacity rates 1654 for a cost of 888. The vehicle cost is greater than the opening cost of hub 2 and the truck capacity is greater than hub 1 and 4 capacity. Such instance characteristics with non Euclidean distances do not favour the GRASP $\times$ ELS solution since hubs and vehicles impose tricky constraints on trip management.

3.5. Implementation and parameters

The two versions of the method (GRASP $\times$ ELS with DFS split procedure and GRASP $\times$ ELS using the greedy split) are implemented using the Borland C++ 6.0 package and experiments were carried out on a quad core 2.83 GHz computer under Windows XP with 8 GB of memory.

The results are matched up to other published methods implemented in Visual C++ and tested on 2.4 GHz PC with 512 MB of RAM and Windows XP. Thus, to have a fair comparative study, the CPU times have been scaled by multiplying the times from the literature by a coefficient 0.8.

Since the GRASP $\times$ ELS is a random search algorithm, each instance is solved five times and the result is the best found solution over the runs with the CPU time required to reach it within the corresponding run.

The following notations are used in the tables:

Cost/Sol:	solution cost
CPU:	computer running time in seconds
BKR:	best-known result
LB:	lower bound
Gap LB:	deviation of a solution value to the lower bound in %
Gap BKR:	deviation of a solution value to BKR in % (negative values indicate improvements)
Avg:	average value

Since the two split procedures do not have the same convergence behaviour, we use the same parameters as proposed in [55], except for the number of GRASP iterations which is set at 625 to the greedy split and 125 for the DFS split.

In addition, N_{max} is a new parameter fixed to 80 000 for both splits (we remember that $N_L = 3$).

3.6. Comparative study on Prodhon's instances

On Prodhon's instances, the proposed two GRASP $\times$ ELS are compared with a lower bound [54] and three published methods: the GRASP of [48], the memetic algorithm with population management (MAPM) of [38] and the cooperative Lagrangean relaxation /granular tabu search (LRGTS) of [49].

Table 2 gives a summary of the overall framework performances of the GRASP $\times$ ELS over the two split procedures. Instances are divided into 4 subsets. Subset one refers to instances with less than 50 customers, subset two with 100 customers and 5 hubs, subset three with 100 customers and 10 hubs and the last one with 200 customers and 10 hubs. The table

Table 2
Overall performances of GRASP × ELS with DFS split.

	GRASP	MAPM	LRGTS	GRASP × ELS greedy split	GRASP × ELS DFS split
$n \leq 50$ and $m=5$ (12 instances)					
Gap BKR	0.53	0.02	0.13	−0.02	**−0.05**
Gap LB	3.71	3.18	3.29	3.14	**3.10**
Best solutions	4	10	5	11	**12**
CPU	1.5	2.4	**0.8**	3.4	9.41
$n=100$ and $m=5$ (6 instances)					
Gap BKR	1.09	0.38	0.30	**−0.29**	−0.23
Gap LB	4.28	3.55	3.47	**2.86**	2.93
Best solutions	0	1	1	**4**	**4**
CPU	22.1	32.5	**4.7**	152.5	135.3
$n=100$ and $m=10$ (6 instances)					
Gap BKR	7.16	3.92	**0.52**	2.84	0.74
Gap LB	17.22	13.62	**9.92**	12.42	10.15
Best solutions	0	1	2	3	**4**
CPU	35.2	30.4	**12.8**	166.7	72.33
$n=200$ and $m=10$ (6 instances)					
Gap BKR	6.20	**0.11**	0.20	0.40	0.34
Best solutions	0	**4**	2	0	1
CPU	422.2	315.8	**68.3**	964.8	814.1
Best solutions overall	4	16	10	**18**	**19**
Gap BKR	3.10	0.89	0.26	0.58	**0.15**
Gap LB	7.2	5.88	4.99	5.39	**4.82**

in the Appendix A gives the results for each instance and each method.

Table 2 proves the GRASP × ELS with the DFS split outperforms the GRASP × ELS using the greedy split procedure. Except for small scale instances, the GRASP × ELS framework based on the DFS procedure is less time consuming with an average CPU time of 72.33 s compared with 166.7 s for the instances with 100 customers and 10 hubs.

In addition, the GRASP × ELS with the DFS split procedure retrieves 19 best solutions and, in comparison with the GRASP × ELS with the greedy split, the deviation from the BKR is reduced from 0.58% to 0.15%. The deviation of 4.82% is the best ever published deviation from the lower bound.

Without any doubt the GRASP × ELS with DFS split procedure outperforms all previous published methods including the GRASP × ELS using the greedy split. Since the difference between the two frameworks comes from the split procedure, one can state that the improvement is due to the DFS.

3.7. *Comparative study on new real-life instances*

Since the new instances are much more difficult to solve, we intend to test whether the new split procedure can still make the GRASP × ELS perform well or not. Thus, we compare its results with those provided by the previously published ones of Duhamel et al. in 2009 using the same GRASP × ELS framework using the greedy split [55].

Table 3 gives a summary of the results by dividing the benchmark into 3 subsets of 10 instances:

- small instances, less than 40 nodes and 100 arcs;
- medium instances, less than 90 nodes and 292 arcs;
- large instances, up to 341 nodes and 1124 arcs.

The conclusion corroborates the fact that the DFS procedure leads to better solutions on average. The main improvement is reported on large scale instances, with a difference close to 5% between the 71 004 average value and the 67 982 average cost reported using the GRASP × ELS based on the new version.

Table 3
Comparative study between the GRASP × ELS with the greedy split and DFS split.

	Avg. cost	Avg. best cpu
Small—greedy split	**9347.05**	**0.50**
Small—DFS split	**9325.65**	**0.10**
Medium—greedy split	**24964.10**	**25.50**
Medium—DFS split	**24882.06**	**14.10**
Large—greedy split	**71004.75**	**205.90**
Large—DFS split	**67982.94**	**525.10**

Both versions of the GRASP × ELS are efficient from a computational time point of view, since small instances are solved in less than 1 s, medium scale instances in less than 20 s and the large scale instances in less than 10 min. However, the GRASP × ELS with the DFS split is from 2 to 5 times faster.

The solution of instance DLPP90 is given in Appendix B to illustrate the particularities of this benchmark. All the solutions related to GRASP × ELS framework based on the greedy split are available at: http://prodhonc.free.fr/ and http://www.isima.fr/~lacomme/lrp/lrp.html.

A full comparative study between the GRASP × ELS framework using the two split versions is provided in Appendix C.

4. Comparative study between the two split procedures for the LRP

As stressed in the previous section, the DFS split significantly increases the performance of the general framework for resource constraint problems. This section aims at comparing in more depth how both split procedures carry out the computation of routes from a given sequence of customers favouring the general framework convergence.

The following experiments are performed:

- Section 4.1 provides a study of the split performance to evaluate given giant tours and the impact of using or not lower and upper bounds to define promising labels;
- Section 4.2 studies the convergence of the DFS split procedure during an evaluation without lower and upper bounds;
- Section 4.3 tries to highlight the GRASP × ELS convergence using the two split procedures without lower and upper bounds.
- Section 4.4 studies the influence of the number of labels stored on each node.
- Finally Section 4.5 provides a comparative study of the two splits with lower and upper bounds computation. The split convergences are evaluated based on the results obtained in Section 4.1.

4.1. Split evaluation

To evaluate only the search efficiency of the split, 20 giant tours were generated and evaluated successively by the two versions of the procedure. Table 4 shows that the DFS:

- outperforms the greedy split 14 times;
- is 20 times more time consuming but investigates 20 times more labels;
- gives more significant improvement in quality solutions for the same set of parameters.

The main difference comes from the number of labels investigated by the two algorithms: the greedy split stops when the last node is fully labelled while the DFS algorithm backtracks. This experiment shows that, for a similar set of parameters, the number of nodes investigated can be quite dissimilar.

Table 4 also provides a comparative study of the two split procedures including both lower and upper bound computation. Evaluation of giant tours shows that computation time decreases from 0.031 to less than 0.0001 for the greedy split and from 0.796 to 0.063 to the DFS split. This leads us to think that including bounds has a significant positive impact on computational time.

4.2. Convergence of split process

In the depth first search split we propose managing an ordered set of labels per node depending on the cost, to put first labels on the last node and then backtracking to improve the solution. This management is close to the typical curve of any branch and bound scheme except that we do not experiment with any branching but instead the more promising unused label. For example, let us consider the first giant tour of the Table 4. The DFS split outperforms the greedy split with a solution of 365 104 compared to the poor solution of the greedy split which values 375 449. The first label saved on the last node value was 429 182 and required only the generation of 456 labels thanks to DFS strategy. Successive split backtracks lead to outperforming the greedy split for the first time with the solution 374 046 at the label generation number 26 616 over the 80 000 permitted. This example of convergence rate, illustrated in Fig. 2, leads us to believe that the DFS has a fine convergence rate.

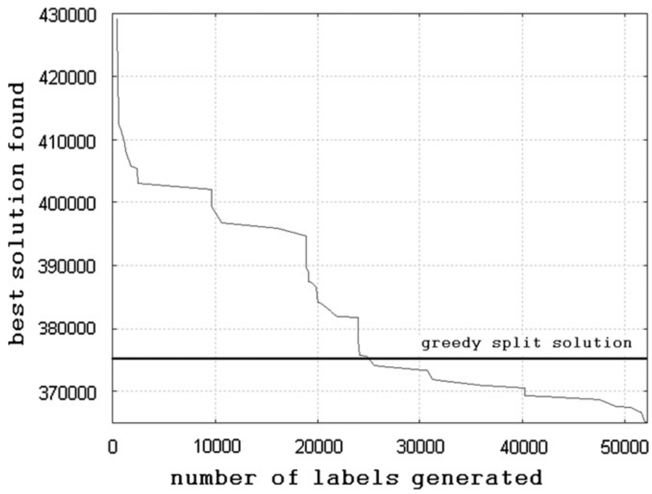

Fig. 2. Depth first search split convergence.

Table 4
Comparative study of split for instance 100-5-1a during the twelve giant tour evaluations with $N_L=3$ and $N_{max}=80\,000$.

Giant trip	Without bounds						With bounds			
	Greedy split			DFS Split			Greedy split		DFS split	
	Sol.	Nb of labels	Cpu	Sol.	Nb of labels	Cpu	Sol.	Cpu	Sol.	Cpu
1	375449	5331	0.031	**365104**	80000	0.796	375449	0.000	**365104**	0.063
2	**388070**	5129	0.047	402672	80000	0.733	388070	0.000	402672	0.047
3	403692	5116	0.031	**388190**	80000	0.812	403692	0.000	**388190**	0.063
4	384223	5007	0.047	**375497**	80000	0.764	384223	0.000	**375497**	0.063
5	412831	5247	0.031	**400150**	80000	0.780	412831	0.000	**400150**	0.062
6	414399	5235	0.047	**393461**	80000	0.811	414399	0.000	**393461**	0.062
7	387722	5249	0.047	**386951**	80000	0.796	387722	0.015	**386951**	0.047
8	404622	5269	0.031	**396657**	80000	0.702	404622	0.016	**396657**	0.031
9	**407355**	5205	0.047	427279	80000	0.718	**407355**	0.000	427279	0.047
10	**407844**	5281	0.031	433617	80000	0.749	**407844**	0.000	433617	0.047
11	398017	5195	0.047	393127	80000	0.796	398017	0.000	**393127**	0.063
12	390845	5287	0.047	381062	80000	0.733	390845	0.016	**381062**	0.047
13	**369109**	5269	0.046	377342	80000	0.780	**369109**	0.000	377342	0.063
14	379083	5288	0.031	368911	80000	0.702	379083	0.000	**368911**	0.047
15	398889	5090	0.032	372006	80000	0.780	398889	0.000	**372006**	0.063
16	390746	5310	0.032	386119	80000	0.670	390746	0.015	**386119**	0.047
17	**398520**	5164	0.031	426400	80000	0.733	**398520**	0.000	426400	0.062
18	385977	5196	0.032	380334	80000	0.733	385977	0.015	**380334**	0.047
19	381555	5122	0.031	356915	80000	0.765	381555	0.000	**356915**	0.062
20	**390217**	5183	0.031	408017	80000	0.733	**390217**	0.016	408017	0.047

This algorithm is a high quality solution procedure which appropriately explores the allowed labels while the greedy split uses only less than 10% of the N_{max} to stop at a weak solution.

Similar comments remain true for almost all the experiments we manage. However, occasionally the DFS cannot compete with the greedy split. This is due to the fact that the DFS split takes advantages of backtracks during label enumeration and thus, depending on the instance and on the node sequence, it may require a significant number of labels to outperform the greedy split. This is stressed in Table 4 where it failed 6 times versus the classical version. Nevertheless, as stressed in Table 2, when applied within a general framework, the results are generally better, and achieved in a shorter period of time. The following section stresses the impact of the giant tour evaluation improvement on the global optimization scheme.

4.3. Influence of split during the global optimization

As previously indicated, once the total number of labels is large enough, the DFS split provides, most of the time, a better evaluation than the greedy with a larger computation time. This evaluation has a positive impact on the global optimization since a giant tour λ can be stated as worse than λ' using the greedy split and better than λ' using the new version.

Due to DFS split, one iteration of the GRASP × ELS is more time consuming but a better solution would appear earlier during the optimization.

Providing a comparative study to highlight the convergence curve is not straightforward since the GRASP × ELS is a stochastic optimization process and the convergence curve heavily depends on replications and on the instance of interest. N_L and N_{max} are clearly two important parameters. Table 5 provides a comparative study on two convergence curves using the instance 50-5-1, where the two split processes retrieve the same final value. Since the DFS provides better evaluation, convergence is better and the GRASP × ELS attains the 90 111 value more quickly.

4.4. Influence of the N_L parameter

If the solution is found achieving the maximal number of labels, then N_{max} may not be large enough especially if N_L is large. However, increasing the number of labels per node could produce weak solution quality. This phenomenon can be highlighted by splitting 20 sequences using instance 100-5-1a with 80 000 labels. The first evaluation of Table 6 gives 365 104 with 3 labels per node and 373 996 with 10 labels per node.

The reason for such behaviour is due to the fact that during a DFS procedure, the algorithm does not immediately develop each generated label on a node. Instead, it stores N_L labels on a stack and tries to reach the last node. Then, a LIFO rule investigates the non-developed labels.

When N_L is large, the number of labels generated before reaching the last node is greater than when N_{max} is small. Later, when backtracking in the sequence, more labels are also investigated since each father is responsible for the generation of $\prod_{p=1}^{t} r_p$ new labels. Thus, the consumption of the N_{max} labels is achieved earlier than for a small N_L value. Thereby the backtracking on labels from nodes at the beginning of the sequence may not be done if N_{max} is too small in comparison with n and N_L. Thus increasing N_L alone may lead to weak solutions.

Table 5
GRASP × ELS convergence curve for instance 50-5-1.

Greedy split with	DFS split
iter=0 best_val=inf	iter=0 best_val=inf
iter=10 best_val=91172	iter=10 best_val=90760
iter=20 best_val=90160	iter=20 best_val=90760
iter=30 best_val=90160	iter=30 best_val=90111
iter=40 best_val=90160	iter=40 best_val=90111
iter=50 best_val=90160	iter=50 best_val=90111
iter=60 best_val=90160	iter=60 best_val=90111
iter=70 best_val=90160	iter=70 best_val=90111
iter=80 best_val=90160	iter=80 best_val=90111
iter=90 best_val=90160	iter=90 best_val=90111
iter=100 best_val=90160	iter=100 best_val=90111
iter=110 best_val=90160	iter=110 best_val=90111
iter=120 best_val=90160	iter=120 best_val=90111
iter=130 best_val=90160	iter=130 best_val=90111
iter=140 best_val=90160	iter=140 best_val=90111
iter=150 best_val=90111	iter=150 best_val=90111
iter=160 best_val=90111	iter=160 best_val=90111
iter=170 best_val=90111	iter=170 best_val=90111
iter=180 best_val=90111	iter=180 best_val=90111
iter=190 best_val=90111	iter=190 best_val=90111

Table 6
Influence of the N_L parameter on the giant tour evaluation ($N_{\max}=80\,000$, instance `100-5-1a`) for the DFS split.

Giant tour	$N_L=3$	$N_L=10$	$N_L=50$
1	**365104**	373996	375449
2	402672	388070	**378647**
3	388190	403692	**387878**
4	**375497**	375**497**	385597
5	**400150**	417675	414215
6	393461	**391583**	391**583**
7	**386951**	387722	387722
8	**396657**	396668	396668
9	427279	**398275**	398**275**
10	433617	408427	**406777**
11	**393127**	393**127**	393**127**
12	**381062**	390845	386498
13	377342	**369017**	369**017**
14	**368911**	387304	383437
15	**372006**	394333	387598
16	386119	**375294**	375**294**
17	426400	**398520**	398**520**
18	380334	385977	**379379**
19	**356915**	371445	371445
20	408017	**390217**	**390217**

Table 7
DFS split performance with $N_{\max}=200\,000$ using different values of N_L.

Giant tour	$N_L=3$	$N_L=10$
1	351151	351151
2	388070	388070
3	370883	370883
4	366871	366871
5	363384	363384
6	395245	395245
7	387722	387722
8	396657	396657
9	366408	366408
10	366305	366305
11	365634	365634
12	363604	363604
13	347671	347671
14	365627	365627
15	353279	353279
16	375294	375294
17	358859	358859
18	364687	364687
19	356941	356941
20	367047	367047

Table 8
Comparative study of split on instance 100-5-1a with N_L=3 and N_{max}=400 000.

Giant trip	With lower and upper bounds				Without lower and upper bounds			
	Greedy split		DFS split		Greedy split		DFS split	
	Sol.	Cpu	Sol.	Cpu	Sol.	Cpu	Sol.	Cpu
1	375449	0.016	351151[+]	0.094	375449	0.016	351151[+]	0.109
2	388070	0.000	**388070**	0.063	388070	0.000	402672[−]	0.063
3	403692	0.016	370883[+]	0.140	403692	0.000	370883[+]	0.156
4	384223	0.016	366871[+]	0.093	384223	0.000	366871[+]	0.109
5	412831	0.000	363384[+]	0.140	412831	0.016	363384[+]	0.140
6	414399	0.016	395245[+]	0.078	414399	0.000	**393461**[+]	0.109
7	387722	0.016	387722	0.093	387722	0.000	**386951**[+]	0.109
8	404622	0.016	396657[+]	0.062	404622	0.000	396657[+]	0.079
9	407355	0.000	366408[+]	0.125	407355	0.000	366408[+]	0.125
10	409331	0.016	360937[+]	0.125	407844	0.000	360588[+]	0.172
11	398017	0.016	365634+	0.094	398017	0.016	365634[+]	0.109
12	390845	0.000	363604[+]	0.078	390845	0.016	**363171**[+]	0.062
13	369017	0.015	347671[+]	0.078	369109	0.000	**347579**[+]	0.125
14	379083	0.016	365627[+]	0.062	379083	0.000	**364698**[+]	0.078
15	399459	0.015	353279[+]	0.094	398889	0.000	**352975**[+]	0.110
16	375294	0.015	**375294**	0.125	390746	0.000	380374[+]	0.125
17	398520	0.016	358859[+]	0.093	398520	0.000	358859[+]	0.125
18	385977	0.016	364687[+]	0.093	385977	0.000	364687[+]	0.093
19	381555	0.016	356941[+]	0.062	381555	0.000	**352873**[+]	0.078
20	390217	0.015	359085[+]	0.125	390217	0.000	359085[+]	0.125

[+] Better result for the DFS procedure than for the greedy one.

It is not surprising that a larger value of N_{max} solves this problem (see Table 7), since the split becomes closer to the optimal version. The split is optimal if and only if there is no limitation on N_{max} and on N_L, which is highly time consuming in a wide majority of problems.

4.5. Lower bound and upper bound consequences on split performances

Thanks to the bounds, some generated labels may be discarded and the algorithm is accelerated. This is particularly true when N_{max} is small (see Table 4), showing the significant impact of the bounds at the beginning of the procedure. Then, as shown in Table 8 with a higher value of N_{max}, the interest is less quantifiable. In addition, discarding non-promising labels gives more chances to develop others (since we still need N_L labels per node). Thus, the final results may sometimes differ from the solutions obtained without using any bounds. The larger the number of generated labels N_{max}, the greater the chance of affecting the final solution. More precisely, the split retrieves equal solutions with N_{max}=80 000 (Table 4) and it is slightly worse with a deterioration of 0.02% with the bounds and N_{max}=400 000 (Table 8). The explanation provided in Section 4.4 for the N_L parameters remains true for this phenomenon. Indeed, the new labels developed may sometimes lead to poor solutions. Interesting values are in bold in Table 8.

Similar observations remain true for the greedy split. For example experiment 10 reports a value of 407 844 for the greedy split without lower and upper bounds and a second value of 409 331 when lower and upper bounds are included.

However, lower and upper bounds in split bring a real advantage in terms of computing time. Thus, when used within a general framework with a reasonable value of N_{max}, the results are encouraging. For instance, the value 90 111 is reached after 30 iterations of the GRASP × ELS applied without bounds on instance `50-5-1` (see Table 5), while it is reached after only 20 iterations when using the lower and upper bounds.

Note also that with N_{max}=400 000, DFS split clearly outperforms the greedy split in term of solution cost.

5. A GRASP × ELS based on split methods for the HVRP

Trying to corroborate the efficiency of the DFS split procedure on a second routing problem, a solution technique based on a *GRASP × ELS* framework is tried out on the heterogeneous vehicle routing problem using the split procedures for swap between giant tours and trips. The well-known classical VRP is not relevant for the split since the shortest path computed on the auxiliary graph would be optimal on both versions (one label per node) and no difference could be highlighted. The HVRP which encompasses several resources is a suitable routing problem for split experiments providing auxiliary graphs with numerous labels per node.

5.1. The heterogeneous vehicle routing problem

In some cases, the classical VRP may not be realistic enough. In fact, when $K>1$ types of vehicles are available, the model can be generalized into a Vehicle Fleet Mix Problem (VFMP) which was first introduced in 1984 by Golden et al. [3]. Each type k has specific capacity Q_k and fixed cost f_k. Choi and Tcha [56] add a cost per distance unit v_k. The goal remains the same as for the VRP except that the total cost of a trip of length $L$is $f_k+L\cdot v_k$. If, in addition, the company has already bought its vehicles, each type k has a limited number a_k of vehicles. From a split point of view, this limitation brings a resource constraint that has to be considered during the computation of the shortest path in the auxiliary graph. This problem is known as the Heterogeneous Fleet VRP (HVRP) and this is the one that will be used in the following experiments.

The HVRP and all the VRP extensions mentioned above are NP-hard since they reduce to the VRP when K=1 and a_k=n. For a review of the literature on VFMP and HVRP, see the recent publication from Prins [28].

5.2. GRASP × ELS framework

As for the LRP, the framework used to test both split procedures is a GRASP × ELS fully described in [57].

5.3. Label definition for split procedures

In both the greedy and DFS split, a label L_i^j is fully defined by $(C, Q_1, \ldots, Q_K, u, v, S_{L_i^j})$ where C is the label cost, Q_i is the number of vehicles of type i remaining available, with k the number of vehicle types. For example $L_i^j = (500, 2, 10, 5 \ldots)$ means that customers $\lambda = (l), \ldots, \lambda = (j)$ can be serviced by a set of trips of cost 500 with 2 vehicles of type 1 remaining free, 10 for vehicles of types 2 and 5 for vehicles of type 3.

The current label L_i^l is used to generate the `final_label` which is inserted if it is promising, implying the check of the two subsequent conditions:

- the current label cost added to a lower bound must be lower than the cost of the upper bound on the last node;
- the biggest residual vehicle in terms of capacity must be enough to serve at least the demand of the largest remaining customer.

This label definition allows the split procedures to handle directly the heterogeneous fleet and the assignment of vehicles to the trips. One can conclude the split addresses efficiently this HVRP key point.

5.4. Benchmarks of interest

5.4.1. Classical HVRP instances

Taillard in 1999 built eight instances based on these VFMP-V files [58] but with limited availabilities a_k of fleets k to represent HVRP. His paper provides results for this kind of problems. Later, Tarantilis et al. [59] designed a threshold accepting algorithm (TA) for the HVRP and used the eight instances from Taillard. Li et al. [61] published a record-to-record (RTR) travel metaheuristic for the HVRP that outperforms the TA. Prins [28] also tackled these instances with his SMA-D2. Table 9 recalls the results from these authors and shows how our two GRASP × ELS perform on this problem.

5.4.2. New HVRP instances (DLP_HVRP)

To provide strongly realistic instances, we use the software introduced by Bajart and Charles [60] for the French counties which provides a set of 96 new instances with scales varying from 20 to 256 nodes and a fleet size varying from 3 to 8 types of vehicles. Shortest paths are computed using the Google web service and represent a true distance in kilometres between cities. The fleet composition has been randomly generated. To the best of our knowledge, these instances are the first real life instances available based on real country districts. They are available at http://www.isima.fr/~lacomme/hvrp/hvrp.html and at http://prodhonc.free.fr/.

5.5. Parameters

Trying to favour fair comparative studies, the set of parameters for the method is defined for all classical and new instances.

The maximal number of labels assigned to the split procedure is set to 50 000 for the two set of instances since the maximal number of labels per nodes is set to 2 for the both sets of instances. The maximal number of iterations assigned to the local search is 100. The number of ELS iterations is 20, the number of GRASP iterations is 50 and finally the number of mutations is set to 15.

5.6. Results on the classical HVRP instances

Since the best published methods obtain results very close to the optimal solutions, it is extremely difficult to distinguish between the influence using either the greedy or the depth first search split. However, it is possible to note, the GRASP × ELS with DFS permits us to reach the 1065.20 and 1120.34 values of, respectively, instances 17 and 19 which are not achieved using the greedy split (see Table 9). This lack of difference can be partially explained because the instances are not large scale ones and have been intensively studied for years: a majority of instances are solved optimally. The results lead us into considering that Depth First Search Split Approach competes with the best published methods including the SMA-D2 method of Prins in 2009.

Table 10
DLP_HVRP instances.

	Average best time	# Best	# Equal to
GRASP with greedy split	552.06	45	8
GRASP with DFS split	518.38	43	8

Table 9
GRASP × ELS performances on HVRP instances with DFS split procedure.

	n	BKS	Taillard		Tarantilis		Li		SMA-D2		GRASP × ELS Greedy Split		GRASP × ELS DFS	
			Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time
13	50	1517.84*	1518.05	473	1519.96	843	**1517.84**	358	**1517.84**	33.20	**1517.84**	90.37	**1517.84**	109.81
14	50	607.53*	615.64	575	611.39	387	**607.53**	141	**607.53**	37.60	**607.53**	25.82	**607.53**	59.49
15	50	1015.29*	1016.86	335	**1015.29**	368	**1015.29**	166	**1015.29**	6.60	**1015.29**	158.97	**1015.29**	6.48
16	50	1144.94*	1154.05	350	1145.52	341	**1144.94**	188	**1144.94**	7.30	**1144.94**	3.17	**1144.94**	47.44
17	75	1061.96*	1071.79	2245	1071.01	363	**1061.96**	216	1065.85	81.50	1065.23	195.64	1065.20	39.42
18	75	1823.58*	1870.16	2876	1846.35	971	**1823.58**	366	**1823.58**	190.60	**1823.58**	60.35	**1823.58**	121.98
19	100	1117.51	**1117.51**	5833	1123.83	428	1120.34	404	1120.34	177.80	1121.06	20.30	1120.34	145.15
20	100	1534.17*	1559.77	3402	1556.35	1156	**1534.17**	447	**1534.17**	223.30	**1534.17**	136.55	**1534.17**	108.56
Avg. dev.			**0.931**		**0.617**		**0.032**		**0.077**		**0.078**		**0.070**	
Avg. time				2011.1		607.1		285.8		94.8		86.40		79.79
Scale time 1.8 GHz				34.7		101.7		213.4		94.8		95.04		87.77
# Best			1		1		7		6		6		6	

Table A1

	GRASP				MAPM				LRGTS				GRASP+ELS				GRASP+ELS DFS Split			
	Cost	Gap LB	Gap BKR	CPU	Cost	Gap LB	Gap BKR	CPU	Cost	Gap LB	Gap BKR	CPU	Cost	Gap LB	Gap BKR	CPU	Cost	Gap LB	Gap BKR	CPU
20-5-1a	55021	0.42	0.42	0.2	54793	**0.00**	**0.00**	0.3	55131	0.62	0.62	0.4	54793	**0.00**	**0.00**	0	54793	**0.00**	**0.00**	0
20-5-1b	39104	**0.00**	**0.00**	0.2	39104	**0.00**	**0.00**	0.3	39104	**0.00**	**0.00**	0.2	39104	**0.00**	**0.00**	0	39104	**0.00**	**0.00**	0
20-5-2a	48908	**0.00**	**0.00**	0.1	48908	**0.00**	**0.00**	0.4	48908	**0.00**	**0.00**	0.5	48908	**0.00**	**0.00**	0	48908	**0.00**	**0.00**	0
20-5-2b	37542	**0.00**	**0.00**	0.2	37542	**0.00**	**0.00**	0.3	37542	**0.00**	**0.00**	0.1	37542	**0.00**	**0.00**	0	37542	**0.00**	**0.00**	0
50-5-1a	90632	6.94	0.52	1.8	90160	6.38	0.00	2.6	90160	6.38	0.00	0.3	90111	**6.32**	**−0.05**	3	90111	**6.32**	**−0.05**	3
50-5-1b	64761	8.71	2.40	1.8	63242	**6.16**	**0.00**	3.2	63256	6.18	0.02	1.0	63242	**6.16**	**0.00**	0	63242	**6.16**	**0.00**	0
50-5-2a	88786	8.20	0.55	2.4	88298	**7.61**	**0.00**	3.4	88715	8.11	0.47	1.8	88643	8.03	0.39	11	88298	**7.61**	**0.00**	28
50-5-2b	68042	6.58	0.51	2.5	67893	6.35	0.29	2.9	67698	6.04	0.00	1.8	67308	**5.43**	**−0.58**	16	67308	**5.43**	**−0.58**	50
50-5-2bis	84055	**2.06**	**0.00**	1.7	84055	**2.06**	**0.00**	3.2	84181	2.22	0.15	2.0	84055	**2.06**	**0.00**	0	84055	**2.06**	**0.00**	5
50-5-2bbis	52059	1.91	0.46	2.6	51822	**1.44**	**0.00**	4.2	51992	1.77	0.33	0.9	51822	**1.44**	**0.00**	11	51822	**1.44**	**0.00**	30
50-5-3a	87380	5.65	1.37	2.3	86203	**4.23**	**0.00**	3.1	86203	**4.23**	**0.00**	0.3	86203	**4.23**	**0.00**	0	86203	**4.23**	**0.00**	0
50-5-3b	61890	4.06	0.10	2.0	61830	**3.96**	**0.00**	4.9	61830	**3.96**	**0.00**	0.5	61830	**3.96**	**0.00**	0	61830	**3.96**	**0.00**	0
Avg.		**3.71**	**0.53**			**3.18**	**0.02**			**3.29**	**0.13**			**3.14**	**−0.02**			**3.10**	**−0.05**	
100-5-1a	279437	2.70	0.54	27.6	281944	3.62	1.44	26.3	277935	2.15	0.00	8.7	276960	**1.79**	**−0.35**	148	277626	2.04	−0.11	321
100-5-1b	216159	4.41	0.59	23.2	216656	4.65	0.82	34.5	214885	**3.79**	**0.00**	8.3	215854	4.26	0.45	68	216457	4.55	0.73	235
100-5-2a	199520	6.74	2.02	17.4	195568	4.63	0.00	35.8	196545	5.15	0.50	2.3	194267	3.93	−0.67	212	194057	**3.82**	**−0.77**	19
100-5-2b	159550	3.72	1.41	22.4	157325	2.27	0.00	36.4	157792	2.58	0.30	3.3	157375	2.31	0.03	125	157195	**2.19**	**−0.08**	45
100-5-3a	203999	5.04	1.12	21.6	201749	3.89	0.00	28.7	201952	3.99	0.10	2.4	200345	**3.16**	**−0.70**	141	200609	3.30	−0.57	184
100-5-3b	154596	3.07	0.83	20.3	153322	2.22	0.00	33.3	154709	3.15	0.90	2.9	152528	1.70	−0.52	221	152473	1.66	−0.55	8
Avg.		**4.28**	**1.09**			**3.55**	**0.38**			**3.47**	**0.30**			**2.86**	**−0.29**			**2.93**	**−0.23**	
100-10-1a	323171	25.14	10.72	37.4	316575	22.59	8.46	24.7	291887	**13.03**	**0.00**	14.1	301418	16.72	3.27	48	299017	15.79	2.44	191
100-10-1b	271477	24.06	15.26	29.5	270251	23.50	14.74	36.0	235532	**7.63**	**0.00**	14.0	269594	23.20	14.46	186	242132	10.65	2.80	126
100-10-2a	254087	11.98	3.66	39.1	245123	8.03	0.00	24.6	246708	8.73	0.65	14.4	243778	**7.44**	**−0.55**	260	244183	7.61	−0.38	16
100-10-2b	206555	6.13	1.04	29.8	205052	5.36	0.30	31.6	204435	5.04	0.00	10.1	203988	**4.81**	**−0.22**	139	203988	**4.81**	**−0.22**	54
100-10-3a	270826	21.80	6.76	35.4	253669	14.08	0.00	29.0	258656	16.33	1.97	13.3	253511	14.01	−0.06	164	253217	**13.88**	**−0.18**	11
100-10-3b	216173	14.19	5.55	39.8	204815	8.19	0.00	36.5	205883	8.76	0.52	10.8	205087	8.33	0.13	203	204801	**8.18**	**−0.01**	36
Avg.		**17.22**	**7.16**			**13.62**	**3.92**			**9.92**	**0.52**			**12.42**	**2.84**			**10.15**	**0.74**	
200-10-1a	490820		1.90	517.5	483497		0.38	345.1	481676		**0.00**	62.0	486467		0.99	1521	485498		0.79	1210
200-10-1b	416753		9.66	379.1	380044		**0.00**	463.0	380613		0.15	60.3	382329		0.60	359	382805		0.73	367
200-10-2a	512679		13.46	554.3	451840		**0.00**	280.6	453353		0.33	60.3	452276		0.10	112	451602		**−0.05**	1279
200-10-2b	379980		1.32	367.4	375019		**0.00**	321.0	377351		0.62	76.9	376027		0.27	1610	376791		0.47	502
200-10-3a	496694		4.20	424.8	478132		0.30	212.9	476684		**0.00**	77.2	478380		0.36	1596	477028		0.07	1442
200-10-3b	389016		6.63	290.2	364834		**0.00**	272.0	365250		0.11	73.3	365166		0.09	591	364836		0.00	85
Avg.			**6.20**				**0.11**				**0.20**				**0.40**				**0.34**	
Global avg.		**7.2**	**3.10**			**5.88**	**0.89**			**4.99**	**0.26**			**5.39**	**0.58**			**4.82**	**0.15**	

5.7. Results on DLP instances

Considering simultaneously the 96 instances, it is difficult to state whether the GRASP × ELS based on the DFS split outperforms the GRASP × ELS based on the greedy split. Table 10 proves the two GRASP × ELS versions have similar average best times with a sight advantage to the DFS (518 s vs. 552 s for the greedy split). However, the GRASP × ELS with greedy split produces 45 best solutions compared with the 43 best ones retrieved by the GRASP × ELS with the DFS split. Appendix D provides the results for the first subset of instances which encompasses instances with less than 100 nodes.

This conclusion must be moderated by a careful analysis of results. Indeed, if both versions provide very similar results on small instances (less than 100 nodes and a gap between the framework using each split procedure equal to 0.02%), that is not true on other sub-sets. The GRASP × ELS based on the DFS split is extremely efficient on instances with between 100 and 150 nodes (better results at 0.7%), and the GRASP × ELS based on the greedy split is the first ranking method for larger instances with 150–200 nodes (better results at 1.1%). Finally, concerning the largest instances with more than 200 nodes, the difference relapses to only 0.2%.

The conclusion is that the quality of the results between both methods is not significantly different on average, but it clearly appears that the choice of the version should be made in relation to the size of the instances.

6. Concluding remarks

This article addresses the split procedures used in routing problems, and provides a new version based on a Depth First Search exploration. Furthermore, a first attempt is proposed to address the challenge in split implementation for "non-classic" optimization problems where labels encompass several costs as well as limited resources.

Experiments on the location-routing problem prove that the DFS split procedure has a great potential since the GRASP × ELS framework using this procedure strongly outperforms the GRASP × ELS based on the greedy split. This is true even for the new instances we introduce, which are supposed to be more difficult to solve since they address non Euclidean distances and a strongly heterogonous set of hubs. The numerical experiments on this new set of instances confirm the preliminary results obtained on Prodhon's instances. Similar concluding remarks hold on the heterogeneous vehicle routing problem for which efficient approaches have been introduced for years.

We also try to bring a deeper analysis to the characteristics of the DFS split taking some Prodhon's instances. It appears that there is no justification for using large values for the maximal number of generated labels N_{max} and for the maximal number of labels stored on the nodes N_L, except if the split optimality is critical in the solution framework. Note that the best results reported have been obtained with very small values for N_L and with reasonable values for N_{max}.

This preliminary study leads us to believe that DFS split is a promising alternative to a standard greedy split procedure. Future research is now directed towards new routing problems where the DFS split could be an efficient way to deal with routing problems such as the heterogeneous LRP, i.e. LRP with heterogonous fleet of vehicles.

Appendix A. Detailed results for Prodhon's instances

See Table A1.

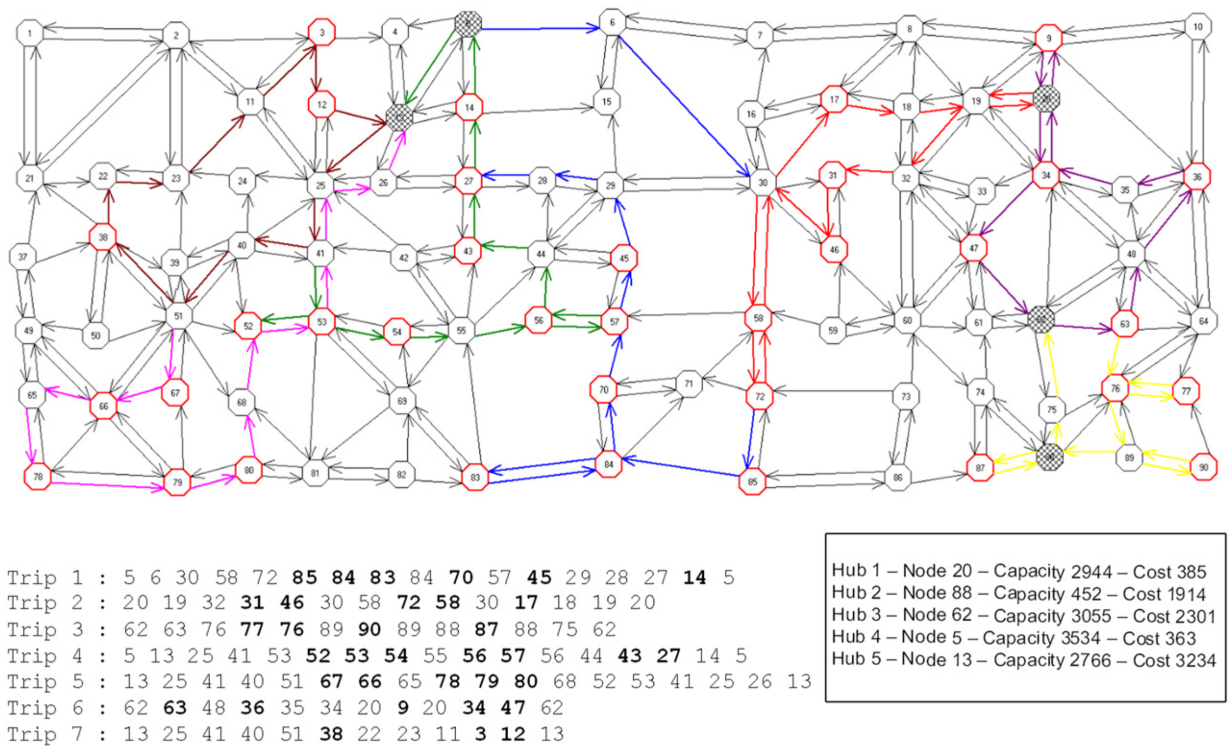

Fig. B1

Table C1

Instance	# Nodes	# Arcs	# Serviced nodes	# Hubs	GRASP × ELS with greedy split (see [55])			GRASP × ELS with depth first search split		
					Value	Best CPU	Total CPU	Value	Best CPU	Total CPU
DLPP_16	16	30	5	2	8163.4	1	71	8163.35	**0**	**0**
DLPP_19	19	52	8	3	8783.0	0	71	8782.99	**0**	**0**
DLPP_21	21	66	10	2	9242.1	1	71	9242.07	**0**	**0**
DLPP_25	25	69	11	2	9255.8	1	72	9255.75	**0**	**1**
DLPP_28	28	76	10	1	15042.5	0	71	15042.50	**0**	**0**
DLPP_32	32	94	15	2	7383.2	0	73	7383.21	**1**	**1**
DLPP_34	34	103	13	2	8018.8	2	72	8018.75	**0**	**1**
DLPP_36	36	92	14	2	9877.2	0	73	**9663.36**	**0**	**1**
DLPP_38	38	112	13	2	6376.1	0	84	6376.11	**0**	**1**
DLPP_40	40	107	104	3	11328.4	0	72	11328.40	**0**	**1**
Avg.					**9347.05**	**0.50**	**73.00**	9325.649	**0.1**	**0.6**
DLPP_42	42	106	18	3	19475.6	1	75	**19345.40**	3	**5**
DLPP_47	47	135	24	3	19029.7	1	72	**18303.00**	7	**11**
DLPP_51	51	138	24	3	20225.7	3	84	**20061.80**	7	**9**
DLPP_52	52	159	29	3	19869.2	4	75	19869.10	**3**	**14**
DLPP_55	55	162	26	5	24978.8	39	81	25089.50	**1**	**18**
DLPP_68	68	228	31	6	19183.2	76	85	19130.60	**19**	**41**
DLPP_76	76	244	30	6	19438.6	45	116	19592.00	**1**	**18**
DLPP_87	87	268	39	5	31626.7	19	125	31787.20	**47**	**48**
DLPP_90	90	292	35	5	24337.9	44	85	24166.40	**30**	**44**
DLPP_100	100	319	50	7	51475.6	23	146	51475.60	**23**	**143**
Avg.					**24964.10**	**25.50**	**94.40**	24882.06	**14.1**	**35.1**
DLPP_110	110	350	49	5	47444.6	106	113	47575.10	50	**107**
DLPP_126	126	403	59	7	35052.4	49	142	35052.40	92	204
DLPP_130	130	401	68	10	36021.6	71	179	36021.60	**22**	464
DLPP_137	137	433	55	6	50233.2	30	125	**49287.40**	45	132
DLPP_170	170	534	72	10	60786.5	154	197	61142.80	234	346
DLPP_210	210	659	100	13	65201.9	11	433	**65094.00**	399	890
DLPP_224	224	710	96	11	81068.6	393	472	81174.10	418	762
DLPP_260	260	863	113	14	89787.1	580	594	89963.60	742	1148
DLPP_285	285	907	130	17	98500.6	133	1017	98746.40	1409	1621
DLPP_341	341	1124	164	23	145951.0	532	2391	**145772.00**	1840	3351
Avg.					**71004.75**	**205.90**	**566.30**	**67982.94**	**525.1**	**902.5**

Table D1

Instance name	**District name**	***n***	**nt**	**TC (kg)**	**TF (kg)**	**GRASP × ELS with classical split**		**GRASP × ELS with DFS split**	
						Cost	**Time**	**Cost**	**Time**
DLP_HVRP_01	Ain	92	4	4000.00	7500.00	**9210.14**	52.29	9219.65	76.23
DLP_HVRP 08	Ardennes	84	3	4555.00	5000.00	**4598.49**	304.85	4600.41	267.35
DLP_HVRP_10	Aube	69	4	3600.00	5350.00	2107.55	24.83	2107.55	75.85
DLP_HVRP_11	Aude	95	4	4800.00	5750.00	**3370.47**	264.61	3373.86	270.71
DLP_HVRP_36	Indre	85	6	1700.00	4650.00	5759.34	104.39	**5752.34**	172.26
DLP_HVRP_39	Jura	77	5	1396.00	2500.00	2934.55	182.11	2934.55	206.60
DLP_HVRP_43	Haute Loire	86	7	6927.00	14450.00	8764.75	219.91	**8762.60**	229.26
DLP_HVRP_52	Haute Marne	59	3	9679.00	21500.00	4029.42	39.97	4029.42	22.83
DLP_HVRP_55	Meuse	56	3	9484.00	11000.00	10247.86	190.76	**10244.34**	136.58
DLP_HVRP_70	Haute Saone	77	4	13755.00	16500.00	**6689.61**	120.60	6708.54	152.69
DLP_HVRP_75	Paris	20	3	700.00	1150.00	452.85	0.02	452.85	0.02
DLP_HVRP_82	Tarn et Garonne	79	3	1900.00	3000.00	4774.26	144.51	**4772.94**	198.88
DLP_HVRP_92	Haut de Seine	35	3	5420.00	22500.00	564.39	20.63	564.39	2.55
DLP_HVRP_93	Seine Saint Denis	40	3	1900.00	3400.00	1036.99	27.39	1036.99	27.66
DLP_HVRP_94	Val de Marne	47	5	3475.00	8250.00	1378.66	15.68	**1378.25**	8.50
Average						4394.62	114.17	4395.91	123.20

Appendix B. Example of solution for instance DLPP90

See Fig. B1.

Appendix C. GRASP × ELS based on the DFS split for the new instances

See Table C1.

Appendix D. GRASP × ELS performances on French districts—DLP_HVRP_1 (instances with less than 100 nodes)

See Table D1.

References

[1] Beasley JE. Route-first cluster-second methods for vehicle routing. Omega 1983;11:403–8.

[2] Haimovich M, Rinnoy Kan A. Bounds and heuristics for capacitated routing problems. Mathematics of Operations Research 1985;10:527–42.
[3] Golden B, Assad A, Levy L, Gheysens F. The fleet size and mix vehicle routing. Computers and Operations Research 1984;11:49–66.
[4] Ulusoy G. The fleet size and mix problem for capacitated arc routing. European Journal of Operational Research 1985;22:329–37.
[5] Ryan DM, Hjorring C, Glover F. Extensions of the petal method for the vehicle routing. Journal of the Operational Research Society 1993;44:289–96.
[6] Renaud J, Boctor FF, Laporte G. An improved petal heuristic for the vehicle routeing problem. Journal of the Operational Research Society 1996;47: 329–36.
[7] Salhi S, Sari M. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. European Journal of Operational Research 1997;103: 95–112.
[8] Lacomme P, Prins C, Ramdane-Cherif W. Competitive genetic algorithms for the Capacitated Arc Routing Problem and its extensions. Lecture Notes in Computer Science 2001;2037:473–83.
[9] Lacomme P, Prins C, Ramdane-Chérif W. Competitive memetic algorithms for arc routing problems. Annals of Operations Research 2004;131:159–85.
[10] Hertz A, Laporte G, Mittaz M. A tabu search heuristic fort the capacitated arc routing problem. Operations Research 2000;48(1):129–35.
[11] Prins C. A simple and effective evolutionary algorithm for the Vehicle Routing Problem. Computers and Operations Research 2004;31(12):1985–2002.
[12] Lacomme P, Prins C, Ramdane-Chérif W. Evolutionary algorithms for periodic arc routing problems. European Journal of Operational Research 2005;165(2): 535–53.
[13] Chu F, Labadi N, Prins C. A scatter search for the periodic capacitated arc routing problem. European Journal of Operational Research 2006;169(2): 586–605.
[14] Lacomme P, Prins C, Sevaux M. A genetic algorithm for a bi-objective capacitated arc routing problem. Computers and Operations Research 2006;33:3473–93.
[15] Fleury G, Lacomme P, Prins C, Ramdane-Cherif W. Improving robustness of solutions to arc routing problems. Journal of the Operational Research Society (JORS) 2005;56:526–38.
[16] Fleury G, Lacomme P, Prins C, Sevaux M. A bi-objective stochastic approach for the stochastic CARP. Research report, Ref. LIMOS RR-08-06, 2008.
[17] Labadi N, Prins C, Reghioui M. GRASP with path relinking for the capacitated arc Routing problem with time windows. Advances in computational intelligence in transport, logistics, and supply chain management. In: Fink A, Rothlauf F, editors. Studies in computational intelligence, vol. 144. Springer; 2008. p. 111–35.
[18] Labadi N, Prins C, Reghioui M. An evolutionary algorithm with distance measure for the split delivery arc routing problem. Recent advances in evolutionary computation for combinatorial optimization. In: Cotta C, van Hemert J, editors. Studies in computational intelligence, vol. 153. Springer; 2008. p. 275–94.
[19] Ngueveu SU, Prins C, Wolfer RC. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. Computers and Operations Research 2010;37:1877–85.
[20] Santos L, Coutinho-Rodrigues J, Current JR. An improved heuristic for the capacitated arc routing problem. Computers and Operations Research 2009;36(9):2632–7.
[21] Corberan A, Prins C. Recent results on arc routing problems: an annotated bibliography. Networks 2010;56:50–69.
[22] Villegas J-G, Medaglia A, Mendoza J, Prins C, Prodhon C, Velasco N. A split-based framework for the vehicle routing problem. CLAIO 2008 (XIV Congreso Latino Ibero Americano de Investigacion de Operaciones), 9–12 September 2008, Catagena, Colombie.
[23] Prins C, Labadi N, Reghioui M. Tour splitting algorithms for vehicle routing problems. International Journal of Production Research 2009;47(2):507–36.
[24] El Fallahi A, Prins C, Wolfler-Calvo R. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Computers and Operations Research 2008;35:1725–41.
[25] Mendoza J, Castanier B, Gueret C, Medaglia AL, Velasco N. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. Computers and Operations Research 2010;37:1886–98.
[26] Labadi N, Prins C, Reghioui M. A memetic algorithm for the vehicle routing problem with time windows. RAIRO. Recherche opérationnelle 2008;42(3): 415–31.
[27] Imran A, Salhi S, Wassan NA. A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. European Journal of Operational Research 2009;197(2):509–18.
[28] Prins C. Two memetic algorithms for heterogeneous fleet vehicle routing problems. Engineering Applications of Artificial Intelligence 2009;22:916–28.
[29] Prins C, Bouchenoua A. A memetic algorithm solving the VRP, the CARP, and more general problems with nodes, edges and arcs. In: Hart W, Smith J, Krasnogor N, editors. Recent advances in memetic algorithms, vol. 166. Berlin/Heidelberg: Springer; 2005. p. 65–85. ISBN: 978-3-540-22904-9.
[30] Boudia M, Prins C, Reghioui M. An effective memetic algorithm with population management for the split-delivery vehicle routing problem. Lecture Notes in Computer Sciences 2007;4771:16–30.
[31] Mosheiov G. Vehicle routing with pick-up and delivery: tour-partitioning heuristics. Computers and Industrial Engineering 1998;34(3):669–84.
[32] Velasco N, Castagliola P, Dejax P, Guéret C, Prins C. A memetic algorithm for a pick-up and delivery problem by helicopter. In: Pereira FB, Tavares J, editors. Bio-inspired algorithms for the vehicle routing problem, Studies in computational intelligence, vol. 161, 2009. p. 173–90.
[33] Duhamel C, Lacomme P, Quilliot A, Toussaint H. A multi-starts evolutionary local search for the two dimensional loading capacitated vehicle routing problem. Computers & Operations Research 2011;38(3):617–40.
[34] Bouly H, Dang D-C, Moukrim A. A memetic algorithm for the team orienteering problem. 4OR—A Quarterly Journal of Operations Research 2009;8(1):49–70.
[35] Lacomme P, Quilliot A, Zhao X. A heuristic and split procedure for dial-a-ride problems. EU/MEeting 2008—Troyes, France, October 23–24.
[36] Nagy G, Salhi S. Nested heuristic methods for the location-routeing problem. Journal of the Operational Research Society 1996;47:1166–74.
[37] Nagy G, Salhi S. The many-to-many location-routing problem. TOP 1998;6(2): 261–75.
[38] Prins C, Prodhon C, Wolfler-Calvo R. A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. In: Lecture Notes in Computer Science, vol. 3906. Berlin: Springer; 2006. p. 183–94.
[39] Duhamel C, Lacomme P, Prins C, Prodhon C. A memetic approach for the capacitated location-routing problem. EU-ME 2008 (EURO Working Group on Metaheuristics), 23–24 October 2008, Troyes.
[40] Duhamel C, Lacomme P, Prins C, Prodhon C. A GRASP $\times$ ELS approach for the capacitated location-routing problem. Computers and Operations Research 2010;37:1912–23.
[41] Prodhon C, Prins C. A memetic algorithm with population management for the periodic location-routing problem. Lecture Notes in Computer Science 2008;5296:43–57.
[42] Villegas JG, Prins C, Prodhon C, Medaglia AL, Velasco N. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. Engineering Applications of Artificial Intelligence 2010;23(5):780–94.
[43] Villegas J-G, Medaglia J, Prins C, Prodhon C, Velasco N. GRASP/evolutionary local search hybrids for a truck and trailer routing problem. In: VIII metaheuristic international conference, Hamburg, July 13–16, 2009.
[44] Desrochers M. An algorithm for the shortest path problem with resource constraints. Research report G-88-27, GERAD, Montreal, Canada, 1988.
[45] Min H, Jayaraman V, Srivastava R. Combined location-routing problems: a synthesis and future research directions. European Journal of Operational Research 1998;108:1–15.
[46] Nagy G, Salhi S. Location-routing: issues, models and methods. European Journal of Operational Research 2007;177:649–72.
[47] Wu TH, Low C, Bai W. Heuristic solutions to multi-depot location-routing problems. Computers and Operations Research 2002;29:1393–415.
[48] Prins C, Prodhon C, Wolfler-Calvo R. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR—A Quarterly Journal of Operations Research 2006;4(3):221–38.
[49] Prins C, Prodhon C, Soriano P, Ruiz A, Wolfler-Calvo R. Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic. Transportation Science 2007;41(4):470–83.
[50] Prins C. A GRASP $\times$ evolutionary local search hybrid for the vehicle routing problem. 2009. In: Pereira FB, Tavares J, editors. Bio-inspired algorithms for the vehicle routing problem, Studies in computational intelligence, vol. 161, 2009. p. 35–53.
[51] Feo TA, Bard J. Flight scheduling and maintenance base planning. Management Science 1989;36:1415–32.
[52] Feo TA, Resende MGC. Greedy randomized adaptive search procedures. Journal of Global Optimization 1995;6:109–33.
[53] Wolf S, Merz P. Evolutionary local search for the super-peer selection problem and the p-hub median problem. Lecture Notes in Computer Science 2007;4771:1–15.
[54] Prodhon C. Le problème de localisation-routage. PhD thesis, University of Technology of Troyes, France, October 2006.
[55] Duhamel C, Lacomme P, Prins C, Prodhon C. A GRASP $\times$ ELS approach for real-life location routing problems. In: International conference on computers and industrial engineering, Université de Technologie de Troyes, 6–8 July, 2009.
[56] Choi E, Tcha D-W. A column generation approach to the heterogeneous fleet vehicle routing. Computers and Operations Research 2007;34:2080–95.
[57] Duhamel C, Lacomme P, Prins C, Prodhon C. A GRASP $\times$ ELS with depth first search split procedure for the HVRP. LIMOS. Research Report LIMOS/RR-10-08, 2010.
[58] Taillard ED. A heuristic column generation method for the heterogeneous fleet VRP. RAIRO Operations Research 1999;33(1):1–14.
[59] Tarantilis CD, Kiranoudis CT, Vassiliadis VS. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. European Journal of Operational Research 2004;152:148–58.
[60] Bajart V, Charles C. Systèmes d'Information Géographique. 3rd year project report, ISIMA. 〈http://www.isima.fr/~lacomme/students.html〉, 2009.
[61] Li F, Golden BL, Wasil EA. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problems. Computers & Operations Research 2007;34:2734–42.

.

Computers & Operations Research 38 (2011) 931–941

Contents lists available at ScienceDirect

Computers & Operations Research

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

A Branch-and-Cut method for the Capacitated Location-Routing Problem

José-Manuel Belenguer [a], Enrique Benavent [a], Christian Prins [b,*], Caroline Prodhon [b], Roberto Wolfler Calvo [c]

[a] Dept. d'Estadística i Investigació Operativa, Universitat de València, Spain
[b] ICD-LOSI (FRE CNRS 2848), Université de Technologie de Troyes, France
[c] LIPN (UMR 7030), Université de Paris Nord, France

ARTICLE INFO

Available online 8 October 2010

Keywords:
Location-Routing Problem
Branch-and-Cut
Vehicle routing
Facility location

ABSTRACT

Recent researches in the design of logistic networks have shown that the overall distribution cost may be excessive if routing decisions are ignored when locating depots. The Location-Routing Problem (LRP) overcomes this drawback by simultaneously tackling location and routing decisions. The aim of this paper is to propose an exact approach based on a Branch-and-Cut algorithm for solving the LRP with capacity constraints on depots and vehicles. The proposed method is based on a zero-one linear model strengthened by new families of valid inequalities. The computational evaluation on three sets of instances (34 instances in total), with 5–10 potential depots and 20–88 customers, shows that 26 instances with five depots are solved to optimality, including all instances with up to 40 customers and three with 50 customers.

1. Introduction

Logistic costs represent an important fraction of company expenses. Therefore, designing an effective and efficient distribution system becomes a strategic question for many industries. Nowadays, even small and medium enterprises should be aware that their future success may depend on the location and distribution decisions underlying the design of their logistic network. These issues are tackled via location-allocation models when customers are served individually by direct (truckload) routes. However, when customers have less-than-truckload demands and thus can be supplied from routes making multiple stops, the assumption of individual routes does not accurately capture the transportation costs. Such situations are better modeled as a Location-Routing Problem (LRP), which integrates depot location and vehicle routing decisions, since a separation often leads to suboptimal solutions as pointed by Salhi and Rand [32].

Several variants of LRP have been addressed in the literature (see Section 2). This paper considers a general form defined on a weighted and undirected network $G=(V,E,C)$. V is a set of nodes comprising a subset I of m possible depot locations and a subset $J=V\backslash I$ of n customers. The case considering any customer as a potential depot location [26] can be modeled without loss of generality, by duplicating the set of customers into I. It is assumed that G is a complete graph from which all edges connecting two depots (nodes in I) are removed. Each edge (i,j) in the edge-set E represents a shortest path in the actual road network, with a given traveling cost c_{ij}. It is assumed that the costs c_{ij} satisfy the triangular inequality. A capacity W_i and an opening cost O_i are associated with each depot site $i \in I$. Each customer $j \in J$ has a demand d_j. A set of identical vehicles of capacity Q is available. When used, each vehicle incurs a fixed cost F and performs one single route.

The LRP consists in opening one or more depots and designing for each opened depot a number of routes whose total customer demand does not exceed the depot capacity. Each route must start and finish at the same depot and the total demand it satisfies must not exceed Q. The total number of vehicles used (or routes performed) is a decision variable. The set of routes must serve all customers and minimize a total cost comprising the fixed costs of open depots, the fixed costs of the used vehicles and the costs of the routes. This problem is NP-hard since it contains as special cases the Capacitated Vehicle Routing Problem or CVRP (single depot case) and the Multi-Depot VRP or MDVRP (case without location decisions). The LRP solution space is even much larger than the one of the CVRP. For instance, consider a CVRP feasible solution with k routes: if m uncapacitated depots are added to get an instance for the LRP, the same routes can be assigned to the depots in m^k distinct ways.

This paper presents a Branch-and-Cut method which solves exactly the LRP or provides good lower bounds. It is organized as follows. The existing literature is recalled in Section 2. Section 3

* Corresponding author. Tel.: +33 325 71 5641; fax: +33 325 71 5649.
E-mail addresses: jose.belenguer@uv.es (J.-M. Belenguer), enrique.benavent@uv.es (E. Benavent), christian.prins@utt.fr (C. Prins), caroline.prodhon@utt.fr (C. Prodhon), roberto.wolfler@lipn.univ-paris13.fr (R.W. Calvo).

0305-0548/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2010.09.019

introduces some required notation and a mathematical formulation of the problem. Some additional constraints are presented in Section 4 to strengthen the formulation. Section 5 is devoted to the Branch-and-Cut algorithm. Computational results are presented in Section 6 and conclusions close the paper.

2. Literature review

The LRP addressed in this paper includes several particular cases studied in the literature. Most of the early papers reported in a survey by Min et al. in 1998 [24] consider either capacitated routes or capacitated depots, but not both; see for instance Laporte et al. [20], Chien [12] and Srivastava [33]. When vehicles incur no capacity constraints, a single route per depot is enough if the triangle inequality holds [2]. Cases with capacitated vehicles involve more decisions: selection of open depots, assignment of customers to these depots, partition of customers assigned to the same depot into routes and sequencing of the customers in each route. The general LRP studied in this paper assumes capacitated depots in addition to capacitated vehicles. Most of the time, the LRP is formulated as a deterministic node routing problem, i.e., customers with known demands are located on nodes of the network. However, a few authors have studied stochastic cases, like Laporte et al. [17], Chan et al. [11] and Albareda-Sambola et al. [3] and, more recently, arc routing versions, see Ghiani and Laporte [14] and Labadi [16].

Solution approaches for the LRP can be divided roughly into heuristics and exact methods, with much of the literature devoted to the first category. Most of the heuristics separate the decision levels into subproblems and handle them sequentially or iteratively. Examples of heuristics can be found in Albareda-Sambola et al. [2] for an LRP with uncapacitated vehicles, and in Tuzun and Burke [35] for an LRP with uncapacitated depots. Concerning the case with capacities on both depots and routes, one can quote Wu et al. [36], Prins et al. [29,30,28], Bruns and Klose [10] and Barreto [8]. The cooperative metaheuristic presented in [28] is on average the most effective on benchmark instances from the literature: it alternates between a location subproblem, solved by Lagrangean relaxation, and a multi-depot VRP, solved by a granular tabu search.

Fewer works have been devoted to the study of exact algorithms for the LRP. Laporte and Nobert [18] designed the first one in 1981, a Branch-and-Bound algorithm for an LRP with a single open depot, and solved instances with up to 50 customers. In Laporte et al. [19], the solution to an LRP with vehicle capacity constraints is obtained by a Branch-and-Cut method. Subtour elimination constraints and chain-barring constraints guarantee that each route starts and ends at the same facility. Computational results are reported for instances with 20 customers and eight depot locations. Laporte et al. [20] addressed an LRP with asymmetrical costs, in which vehicle capacity is replaced by a maximum route length. They elaborated a Branch-and-Bound algorithm able to solve instances with up to 40 customers, but the number of depots is small (2 or 3) and the number of routes per open depot is limited to 2.

Lately, Albareda-Sambola et al. [2] also studied a lower bound for the LRP with uncapacitated routes (one route per depot). In Barreto's thesis [8], a different lower bound is obtained for the general LRP by a cutting plane scheme in which an integer linear program is solved at each iteration. The first set-partitioning formulation of the LRP and a column generation approach were recently introduced by Akca et al. [1], who solved to optimality instances with up to 40 customers. In a conference paper [9], Belenguer et al. proposed two different formulations for the general LRP and a preliminary Branch-and-Cut algorithm able to solve 11 instances with up to 32 customers. In the present paper, additional constraints are introduced to considerably strengthen the second formulation. Moreover, several separation algorithms are implemented, leading to a much more efficient algorithm since all instances with five potential facilities and up to 40 customers are solved to optimality.

3. Mathematical formulation

The LRP model developed here is an integer linear program that has many points in common with the one proposed by Laporte et al. [19], although our formulation includes new constraints to take the capacity of the depots into account and uses binary variables only. The depot variables y_i are equal to 1 if depot i is opened and to 0 otherwise. The edge variables x_{ij} are equal to 1 only if edge $(i,j) \in E$ is traversed exactly once in the solution. Note that there are no variables x_{ij} with $i,j \in I$ since such edges are excluded from the input network. Finally, variables w_{ij} with $j \in J$ and $i \in I$ are used to model trips with only one client, called *return trips* in the sequel. They are equal to 1 if a vehicle uses edge (i,j) twice (the vehicle serves only customer j, from depot i) and to 0 otherwise.

If $i \in I$ and $j \in J$, note that $x_{ij}=1$ implies $w_{ij}=0$ and $w_{ij}=1$ implies $x_{ij}=0$. The following notation is also used:

$x(H) = \sum_{(i,j) \in H} x_{ij}$, $\forall H \subseteq E$, and similarly for $w(H)$,
$\delta(S)$, $\forall S \subseteq V$, denotes the set of edges with one end-node in S and the other in $V \backslash S$,
$\gamma(S)$, $\forall S \subseteq V$, denotes the set of edges with both end-nodes in S,
$(S:S')$, $\forall S \subseteq V$ and $\forall S' \subseteq V \backslash S$, denotes the set of edges with one end-node in S and the other in S'.
$x((S:S'))$ and $w((S:S'))$ will be simplified to $x(S:S')$ and $w(S:S')$, respectively.
$D(S) = \sum_{j \in S} d_j$, $\forall S \subseteq J$, is the total demand in S while $k(S) = \lceil D(S)/Q \rceil$, $\forall S \subseteq J$, is a lower bound to the minimum number of vehicles to serve S.

Thus, the LRP can be formulated as follows:

$$\min \sum_{(ij) \in E} c_{ij} x_{ij} + \sum_{i \in I} \sum_{j \in J} 2 c_{ij} w_{ij} + \frac{F}{2} \sum_{i \in I} \sum_{j \in J} (x_{ij} + 2 w_{ij}) + \sum_{i \in I} O_i y_i \quad (1)$$

s.t.

$$2w(\{j\}:I) + x(\delta(j)) = 2 \quad \forall j \in J \quad (2)$$

$$x(\gamma(S)) \le |S| - k(S) \quad \forall S \subseteq J \quad (3)$$

$$x(S:J \backslash S) + x(S:I \backslash \{i\}) + 2w(S:I \backslash \{i\}) \ge 2 \quad \forall S \subseteq J,\ \forall i \in I \quad \text{such that } D(S) > W_i \quad (4)$$

$$x_{ij} + w_{ij} \le y_i \quad \forall i \in I,\ \forall j \in J \quad (5)$$

$$x(\gamma(S \cup \{l,j\})) + x(\{j\}:I') + x(\{l\}:I \backslash I') \le |S| + 2 \quad \forall j,\ l \in J,\ \forall S \subseteq J \backslash \{l,j\},\ \forall I' \subset I \quad (6)$$

$$x(\{j\}:I) + w(\{j\}:I) \le 1 \quad \forall j \in J \quad (7)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in E \quad (8)$$

$$w_{ij} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J \quad (9)$$

$$y_i \in \{0,1\} \quad \forall i \in I \quad (10)$$

The objective function (1) represents the total cost which is defined as the sum of the fixed costs of open depots, the fixed cost of vehicles used and the total cost of the routes. Constraints (2) are the degree constraints for client nodes. Constraints (3) are the capacity constraints for the vehicles, which are well known and used for the CVRP. Note that at least $k(S)$ vehicles are needed to serve all the clients in S, so the number of edges between nodes in S must be, at most, $|S|-k(S)$. These constraints are also called subtour elimination constraints when $k(S)=1$.

Constraints (4) impose the capacity restriction for the depots: if $D(S)>W_i$, then the clients in S cannot be completely served from depot i and, therefore, there must exist at least two links from nodes in S to a different depot or to another client. In constraints (5), the edges incident with a depot may be used only if this depot is open. Note that $x_{ij}+w_{ij}\le 1$ always holds because variables x_{ij} and w_{ij} cannot be simultaneously equal to 1.

Constraints (6) are called path elimination constraints and they prevent solutions that include a path beginning at one depot and ending at another depot. They are inspired by the chain-barring constraints introduced by Laporte et al. [19] and are explained as follows. Note that $x(\{j\}:I')+x(\{l\}:I\backslash I')\le 2$ and $x(\gamma(S\cup\{l,j\}))\le |S|+1$ because of the subtour elimination constraints. If this last inequality is satisfied with equality, then all the clients in $S\cup\{l,j\}$ are consecutive in a path, but then if, for instance $x(\{j\}:I')=1$ holds, it means that the path is connected to a depot in I' so it cannot be also connected to a depot in $I\backslash I'$, and $x(\{l\}:I\backslash I')=0$ holds. These constraints are also valid in the case where $S=\emptyset$ but they do not prevent an illegal solution where a route starts at one depot, serves a single client and ends at another depot. Nevertheless, this kind of solutions is eliminated by constraints (7). Finally, constraints (8)–(10) are the integrality constraints.

Note that the families of constraints (3), (4) and (6) contain a number of constraints that grows exponentially with the size of $|J|$. In order to use this formulation to solve efficiently the LRP, we use a cutting plane scheme to generate these inequalities only when needed, i.e., in addition to relaxing the integrality constraints, we relax constraints (3), (4) and (6) to get a valid lower bound for the LRP and only those inequalities that are violated by the LP solution are actually added to the LP.

4. Additional constraints

In this section, we introduce several new families of constraints that have proved to be useful in order to strengthen the LP relaxation of the above formulation. Some of them are just improved versions of the constraints which are already present in the formulation. An interesting question is if these constraints induce facets of the corresponding polyhedron, in which case, they would be “best” in some sense. Unfortunately this study is extremely difficult and is not addressed here. Nevertheless, we have tried to improve the constraints in this spirit, trying to use constraints that, as far as we know, are not dominated by other constraints. One technique we have used to strengthen several kinds of constraints is to lift them by including some w-variables. These lifted inequalities have proved to be very effective to remove fractional solutions in which many w-variables with small values appeared.

Some of the constraints introduced below are not valid for all LRP solutions but are satisfied by at least one optimal solution; they are commonly named *optimality cuts*. Given that we assume that the cost matrix satisfies the triangular inequality, it is easy to see that there always exist optimal solutions of the LRP that *do not contain two or more routes based at the same depot and jointly serving a total demand not greater than Q*. We will refer to this property as the TI-property.

In what follows the inequalities that are valid for all feasible solutions are presented prior to the inequalities that are valid only for solutions satisfying the TI-property.

4.1. Constraints derived from the CVRP

The Capacitated Vehicle Routing Problem can be considered as a special case of the LRP with only one depot and, consequently, without depot opening costs and capacities. The CVRP has been widely studied and there exists a vast literature on it, including both heuristics and exact methods. See for instance [25,34,31,15] for some comprehensive surveys on the recent advances on the CVRP.

Concerning the LP based methods, Lysgaard et al. [23] presented a Branch-and-Cut method to solve the CVRP, which makes use of several classes of valid inequalities: capacity, framed capacity, generalized capacity, strengthened comb, multistar and hypotour. Let d be the depot node in the CVRP and J the set of clients, as in the LRP. The CVRP is formulated in [23] using two-index variables, say z_{ij}, that determine the edges included in the solution. More precisely, for one edge (i,j) between two customers, z_{ij} is equal to 1 if the edge is traversed by a vehicle and to 0 otherwise, while for one edge (d,j) incident to the depot z_{dj} is equal to the number of times the edge is used (twice for a return trip). These variables are very similar to the ones we have used to formulate the LRP and we can take advantage of this fact to derive valid constraints for our formulation from any CVRP constraint.

Given any LRP solution, if all depots are shrunk into one single depot, a CVRP feasible solution is obtained. The relationship between the variable values in the CVRP and the x and w variables of our formulation is as follows: $z_{ij}=x_{ij}$ for edges $(i,j), i,j\in J$, and, $z_{dj}=\sum_{i\in I}(x_{ij}+2w_{ij})$, for the edges $(d,j), j\in J$. Therefore, any valid constraint for the CVRP is still valid for all these LRP solutions and can be used to derive a valid constraint for the LRP. Specifically, any valid inequality for the CVRP can be written as

$$\sum_{j\in J}\alpha_{dj}z_{dj}+\sum_{i,j\in J}\beta_{ij}z_{ij}\le f$$

Then, the following is a valid constraint in our LRP formulation:

$$\sum_{j\in J}\sum_{i\in I}\alpha_{dj}(x_{ij}+2w_{ij})+\sum_{i,j\in J}\beta_{ij}x_{ij}\le f$$

Some of the valid constraints for the LRP derived in this way can be strengthened by considering the special features of this problem. For instance, capacity constraints for the CVRP correspond to capacity constraints (3), but these constraints can be strengthened leading to *WCAP* constraints (15) that will be introduced in Section 4.5. Nevertheless, we have not addressed this task for other constraints derived from the CVRP and we have used them just as they result from the above transformation. Trying to reinforce these constraints for the LRP is indeed an interesting research topic to be addressed in the near future.

4.2. Improved depot capacity constraints

Depot capacity constraints (4) can be strengthened by including some w-variables. Furthermore, they can be generalized to deal with a subset of depots instead of only one.

$$x(S:J\backslash S)+x(S:I\backslash I')+2w(S\backslash S':I\backslash I')\ge 2$$
$$\forall S\subseteq J, S'\subset S, \forall I'\subset I \quad \text{such that } D(S\backslash S')>\sum_{i\in I'}W_i \tag{11}$$

Note that if $w(S\backslash S':I\backslash I')=0$ (no client in $S\backslash S'$ is served by a return trip from a depot in $I\backslash I'$), then, given that $D(S\backslash S')>\sum_{i\in I'}W_i$ holds, at least two edges must connect clients in S to clients outside S and/or to depots not in I'. On the other hand if $w(S\backslash S' : I\backslash I') \geq 1$, then (11) is obviously satisfied.

Depot capacity constraints can also be strengthened using a different idea. Given a subset of clients S and a subset of depots I', we denote $r(S,I') = \max\{0, \lceil (D(S) - \sum_{i \in I'} W_i)/Q \rceil\}$. Consider the following inequalities:

$$x(S : J\backslash S) + x(S : I\backslash I') + 2w(S : I\backslash I') \geq 2r(S,I') + 2(1-y_i)(r(S,I'\backslash\{i\}) - r(S,I')), \forall S \subseteq J \quad \forall I' \subset I,\ \forall i \in I' \tag{12}$$

Note that $r(S,I')$ is a lower bound to the minimum number of vehicles needed to serve the demand of S that cannot be served from the depots in I'. Moreover, the right hand side (RHS) of (12) is equal to $2r(S,I')$ if $y_i = 1$, and equal to $2r(S,I'\backslash\{i\})$ if $y_i = 0$. Hence, constraints (12) are valid.

4.3. Improved path elimination constraints

Path elimination constraints (6) can be strengthened in a similar way:

$$w(S \cup \{l,j\} : I) + x(\gamma(S \cup \{l,j\})) + x(\{j\} : I') + x(\{l\} : I\backslash I') \leq |S| + 2 \quad \forall j,l \in J, \forall S \subseteq J\backslash\{l,j\},\ \forall I' \subset I \tag{13}$$

Let us explain the validity of (13). Given an LRP solution, let $p, 0 \leq p \leq |S|+2$, be the number of clients from $S \cup \{l,j\}$ that are served with return trips, then $w(S \cup \{l,j\} : I) = p$. If $p = |S|+2$, the inequality is obviously satisfied because the other terms in the LHS will be zero. If $p < |S|+2$, then $x(\gamma(S \cup \{l,j\})) \leq |S|+1-p$ because no subtour can exist among the remaining $|S|+2-p$ clients. Let us first consider the case where $x(\gamma(S \cup \{l,j\})) = |S|+1-p$. In this case, there is a path containing the $|S|+2-p$ clients that are not served with return trips, but then $x(\{j\} : I')$ and $x(\{l\} : I\backslash I')$ cannot be simultaneously equal to one because, otherwise, we would get a route starting and ending at two different depots, which is not allowed. Finally, note that if $x(\gamma(S \cup \{l,j\})) \leq |S|-p$, given that $x(\{j\} : I') + x(\{l\} : I\backslash I') \leq 2$, inequality (13) is also satisfied.

Fig. 1 shows an infeasible integer solution that violates this constraint where $w(S \cup \{l,j\} : I) = p = 2$, while Fig. 2 represents an LRP solution which satisfies (13).

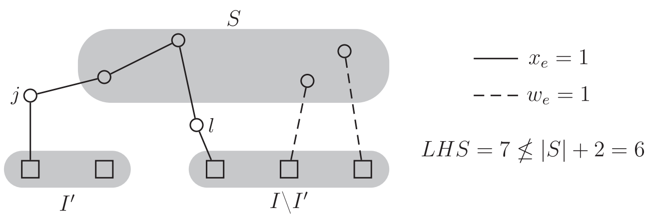

Fig. 1. Violated improved path elimination constraint.

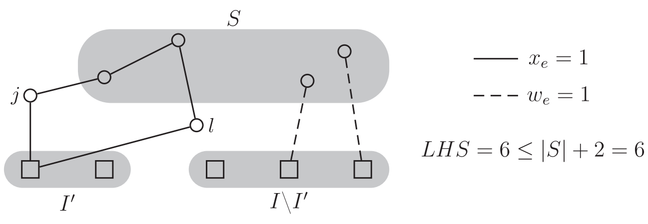

Fig. 2. Solution that satisfies the improved path elimination constraint.

4.4. Co-circuit constraints

Given any LRP solution, if we remove the links that correspond to w-variables (return trips) the number of edges used by the solution in any edge cut-set must be even because it is composed of a set of circuits. This observation leads to the following constraints that are usually called co-circuit inequalities [7]:

$$x(\delta(S)\backslash F) \geq x(F) - |F| + 1 \quad \forall S \subseteq J \cup I, \quad \forall F \subseteq \delta(S), |F| \text{ odd} \tag{14}$$

If $x(F) < |F|$, constraints (14) are trivially satisfied because the RHS is non-positive. In the case that $x(F) = |F|$, given that $|F|$ is odd, at least one edge from $\delta(S)\backslash F$ must be used and thus the constraint is satisfied.

4.5. WCAP constraints

Capacity constraints (3) can also be lifted by including some w-variables. This technique has produced two classes of constraints, both of them will be called WCAP constraints. The first class is

$$w(S' : I) + x(\gamma(S)) \leq |S| - k(S\backslash S') \quad \forall S \subseteq J \quad \text{and} \quad S' \subset S \tag{15}$$

One can see that (15) includes the capacity constraints (3) for $S' = \emptyset$. On the other hand if $S' \neq \emptyset$ and $k(S\backslash S') = k(S)$, then (15) is stronger than (3) because of the term $w(S' : I)$. The validity of these constraints can be proved as follows. Consider any feasible solution to the LRP and let $S'' \subseteq S'$ be the clients of S' that are served with a return trip from any depot, then $|S''| = w(S' : I)$. At least $k(S\backslash S'')$ vehicles are needed to serve the clients in $S\backslash S''$, which cannot be linked to clients in S'' (already served by a return trip). Hence, we have: $x(\gamma(S)) = x(\gamma(S\backslash S'')) \leq |S| - |S''| - k(S\backslash S'') \leq |S| - |S''| - k(S\backslash S')$ and the inequality holds.

Fig. 3 shows an LRP solution satisfying this constraint (the numbers correspond to demands). Fig. 4 represents a fractional solution for the same instance that violates this constraint; it is easy to see that this solution satisfies all the original capacity constraints (3).

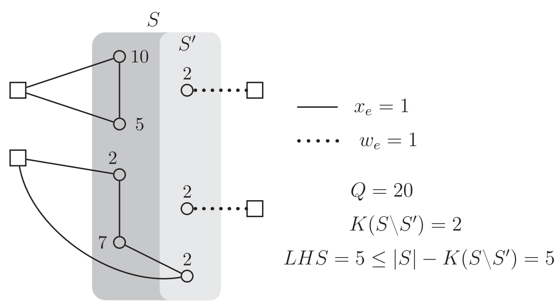

Fig. 3. LRP solution satisfying (15).

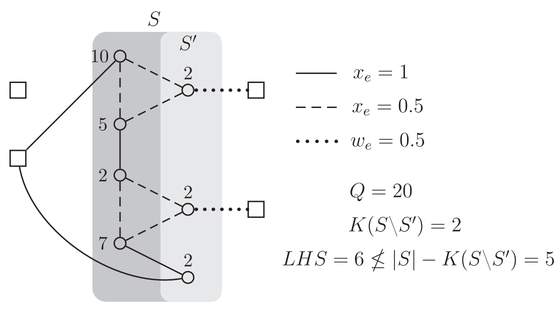

Fig. 4. Fractional solution that violates (15).

The particular case of (15) where $D(S) \leq Q$ corresponds to what we call w-subtour elimination constraints:

$$w(S':I)+x(\gamma(S)) \leq |S|-1 \quad \forall S \subseteq J,\ D(S) \leq Q \quad \text{and} \quad S' \subset S, |S'| = |S|-1 \tag{16}$$

The second class of WCAP constraints are in fact optimality cuts because they are valid only for feasible solutions satisfying the TI-property:

$$w(S:I')+x(\gamma(S)) \leq |S|-1 \quad \forall S \subseteq J \quad \text{such that } d_j+d_l \leq Q \quad \forall j,l \in S \text{ and } \quad \forall I' \subseteq I,\ |I'| \leq |S|-1 \tag{17}$$

Consider any feasible solution to the LRP satisfying the TI-property and let $S'' \subseteq S$ be the clients of S that are served with a return trip from any depot in I', then $|S''| = w(S:I')$. Given that the TI-property holds and $d_j+d_l \leq Q, \forall j,l \in S$, this solution cannot include more than one return trip based at the same depot, so $|S''| \leq |I'| \leq |S|-1$ which implies that $S'' \neq S$. Therefore, $x(\gamma(S)) = x(\gamma(S \backslash S'')) \leq |S|-|S''|-1$, and the inequality holds.

4.6. Depot degree constraints

The following constraints are optimality cuts since they are only valid for solutions satisfying the TI-property. This property allows us to limit in some way the number of routes based at the same depot. For instance, the following constraints bound the number of return trips that can be based at the same depot:

$$w(S:\{i\}) \leq y_i \quad \forall S \subseteq J \quad \text{such that } d_j+d_l \leq Q, \quad \forall j,\ l \in S,\ \forall i \in I \tag{18}$$

Constraints (18) are valid for solutions satisfying the TI-property because, given that any pair of clients of S can fit in one route, two return trips can always be merged into one feasible route. The same idea can be applied to routes with more than one client, leading to the following constraints:

$$2w(S:\{i\})+x(S:\{i\})+x(\gamma(S)) \leq 2y_i+|S|-1 \quad \forall S \subseteq J,\ D(S) \leq Q, \quad \forall i \in I \tag{19}$$

Note that (19) is obviously satisfied when $y_i=0$ because, in this case, $2w(S{:}\{i\})+x(S{:}\{i\})$ will be zero and $x(\gamma(S)) \leq |S|-1$ always holds. On the other hand, due to the TI-property, if $y_i=1$ at most one route may exist based on depot i and serving only clients from the set S; therefore $2w(S:\{i\})+x(\gamma(S))+x(S:\{i\}) \leq |S|+1$ because at most one route can be completely contained in $S \cup \{i\}$.

5. Branch-and-Cut algorithm

The LRP formulation and the additional constraints described in Section 4 have been embedded in a cutting plane scheme to produce a valid lower bound (LB) for the LRP. This cutting plane is the core of the Branch-and-Cut algorithm we propose to solve the LRP. The cutting plane scheme works as follows. Initially, we build a linear program (LP) including the objective function and a subset of constraints to be described later. Then, at each iteration, we solve the current LP, we look for a set of valid inequalities that are violated by the optimal LP solution, we add them to the LP and proceed as before. The cutting plane algorithm stops when no violated inequality is found.

5.1. Initial relaxation

The initial linear program includes the objective function, the degree constraints for the client nodes (2), constraints (5) and (7), the bounds on the variables and some additional inequalities that have been added to strengthen the initial LB. Some of those additional inequalities are special cases of the constraints described in the previous section:

- The capacity constraint (3) with $S = J$.
- The improved depot capacity constraints (12) with $S = J$ and $I' = \{i\}$, for each $i \in I$.
- The depot degree constraints (18) with $S = \{j \in J : d_j \leq Q/2\}$ and each $i \in I$.

Furthermore, the following inequalities are also included in the initial linear program:

$$\sum_{j \in J}(x_{ij}+2w_{ij}) \geq 2y_i \quad \forall i \in I \tag{20}$$

$$\sum_{i \in I} W_i y_i \geq D(J) \tag{21}$$

$$\sum_{i \in I} y_i \geq \min\left\{|I'| : I' \subseteq I \text{ and } \sum_{i \in I'} W_i \geq D(J)\right\} \tag{22}$$

$$\sum_{i \in I}\left\lceil \frac{W_i}{q} \right\rceil y_i \geq \left\lceil \frac{D(J)}{q} \right\rceil \quad \text{for some integers } q>0 \tag{23}$$

Constraints (20) guarantee that each opened depot is really used. Constraint (21) assures that the total capacity of the opened depots is enough to cover the total demand of the clients. Constraint (21) is equivalent to a knapsack inequality if y-variables are replaced by their complements. Constraint (22) uses a lower bound on the minimum number of depots that have to be opened and it can be considered as a special case of the extended cover constraints that are valid for the knapsack polytope associated to inequality (21). In fact we tried to use other valid constraints derived from the knapsack inequality (21), apart from (22), but they were finally discarded because of a negligible impact.

Constraints (23) are Chvàtal–Gomory integer cuts that were also derived from inequality (21) for different integer numbers $q>0$. Note that if $q_1 < q_2$ and $\lceil W_i/q_1 \rceil = \lceil W_i/q_2 \rceil$ for all $i \in I$, then the constraint (23) for q_2 is dominated by the one for q_1. The values of q used are those producing non-dominated constraints among the following ones: $\{W_i : i \in I\}$ and $\{2\alpha, \ldots, 10\alpha\}$, where α is the greatest common divisor of the integers W_i for $i \in I$.

5.2. Separation algorithms

Several algorithms have been applied to find valid inequalities violated by the current LP solution. Some of them are the adaptation to our problem of the heuristics used in the context of the CVRP, since some kinds of inequalities are similar. Additionally, we have designed new procedures for the inequalities that are specific to the LRP: depot capacity constraints (11) and (12), depot degree constraints (19) and path elimination constraints (13). Most of the procedures used are heuristic algorithms but we also implemented exact methods to separate path elimination, co-circuit and w-subtour constraints.

Let $(\overline{x},\overline{w},\overline{y})$ be the optimal solution of the current LP. We denote by $G[\overline{x},\overline{w}]$ the weighted graph induced by the edges e with positive weight that is defined as $\overline{x}_e+2\overline{w}_e$ if $e \in \delta(I)$ and $\overline{x}_e$ if $e \in E \backslash \delta(I)$. This will be called the support graph.

5.2.1. Separation of WCAP constraints

The w-subtour constraints (16) are a special case of WCAP constraints (15) that can be separated in polynomial time (see Subsection 5.2.2). We have also implemented heuristic algorithms for the general case that use three procedures proposed in Augerat et al. [6,5] to separate capacity constraints in the CVRP. These procedures are used to generate tentative sets of customers S, for which the most appropriate subsets S' and I' are then chosen in a greedy way with the objective of finding most violated WCAP constraints. The first procedure simply computes the connected components of the support graph and of the support graph without the depots. The second is based on the idea of sequentially shrinking edges of weight equal to one in the support graph. The third procedure is a tabu search algorithm that works on the shrunk support graph [6,5]. Finally, the procedures implemented in Lysgaard's library [22] to detect violated capacity constraints are also included. The interested reader is referred to the respective papers for additional details.

We describe now the procedure used to find subsets $S' \subseteq S$ and $I' \subseteq I$ from any subset $S \subseteq J$ provided by these heuristics. Given that WCAP constraints represent an improvement of capacity constraints, this procedure is applied to every subset $S \subseteq J$ satisfying $\overline{x}(\gamma(S))+0.25 > |S|-k(S)$ in the hope that adding some w-variables to the LHS would make the WCAP constraint to be effectively violated. The procedure distinguishes two cases:

(a) $D(S) \le Q$. If $|I| \le |S|-1$, then take $I' = I$ and check the violation of (17). Otherwise, build $I' = \{i \in I : \sum_{j \in S} \overline{w}_{ij} > 0\}$. If $|I'| \le |S|-1$, check (17) too. Otherwise, let j^* be the client included in S for which $\sum_{i \in I} \overline{w}_{ij}$ is minimum: take $S' = S \backslash \{j^*\}$ and check the violation of (16).

(b) $D(S) > Q$. Sort the clients $j \in S$ in non-increasing order of $\overline{w}(j) = \sum_{i \in I} \overline{w}_{ij}$, then build set S' by selecting them in this order while $k(S \backslash S') = k(S)$ is satisfied, until no more client can be added to S'. Finally, check the violation of (15) for this S'.

5.2.2. An exact procedure to separate w-subtour elimination constraints

It is well known that subtour elimination constraints are valid in the two-index formulation of the TSP and the CVRP and they can be separated in polynomial time (see for instance [27,5]) by finding a minimum cut in certain support graph. In this subsection we show that this procedure can be adapted to separate w-subtour elimination constraints (16) in polynomial time. Constraints (16) are defined by a set $S \subseteq J$ and a subset $S' \subset S$ with $|S'| = |S|-1$. Let us denote by u to the unique client in $S \backslash S'$; then it is easy to see that, using the degree equations, inequality (16) can also be written as

$$x(\delta(S))+2w(\{u\} : I) \ge 2 \quad \forall S \subseteq J \quad \text{and} \quad u \in S,\ |S| \ge 2 \tag{24}$$

Let $G[\overline{x}]$ be the graph induced by the edges e with positive $\overline{x}_e$. Form a graph G' by adding to $G[\overline{x}]$ a sink node, say t, connected to all depots with infinite weight edges. Then, for every client $u \in J$, compute the minimum weighted cut separating node u from the sink node t. If the weight of this cut is less than $2-2w(\{u\}{:}I)$ the w-subtour inequality (24), where set S is the set of nodes defining that cut in the shore of u, is violated; otherwise no such inequality for client u is violated.

5.2.3. An exact procedure to separate path elimination constraints

We will assume that constraints (7) and w-subtour elimination constraints are satisfied by the current LP solution $(\overline{x},\overline{w})$ when searching for a violated path elimination constraint (13) (also called path constraint for short).

Note that path constraints are determined by a pair of clients $\{j,l\}$, a subset of clients $S \subseteq J \backslash \{j,l\}$ and a subset of depots $I' \subset I$. In what follows we describe a procedure that has to be applied to every pair of clients $\{j, l\}$. It is obvious that (13) will never be violated if j and l belong to different connected components of the support graph $G[\overline{x},\overline{w}]$, so we only consider pairs of clients $\{j, l\}$ belonging to the same connected component.

Given the pair of clients $\{j, l\}$, we observe that the LHS of (13) can be divided in two parts. The first part, $\overline{x}(\{j\} : I')+\overline{x}(\{l\} : I \backslash I')$, that depends only on the selection of I', and the second part, $\overline{w}(S \cup \{l,j\} : I)+\overline{x}(\gamma(S \cup \{l,j\}))$, that depends only on the selection of S. The first part reaches its maximum value for $I' = \{i \in I : \overline{x}_{ij} \ge \overline{x}_{il}\}$. It can be shown that if $I' = I$ or $I' = \emptyset$ no path constraint is violated for the given pair $\{j, l\}$; otherwise, we denote $A(j,l,I') = \overline{x}(\{j\} : I')+\overline{x}(\{l\} : I \backslash I')$. Let us now show how to compute S to maximize the second part of the LHS.

Note that adding the degree constraints for all clients in $S \cup \{l,j\}$, we obtain

$$2(|S|+2) = 2\overline{x}(\gamma(S \cup \{l,j\}))+2\overline{w}(S \cup \{l,j\} : I)+\overline{x}(\delta(S \cup \{l,j\}))$$

Therefore, maximizing $\overline{x}(\gamma(S \cup \{l,j\}))+\overline{w}(S \cup \{l,j\} : I)$ is equivalent to minimizing $\overline{x}(\delta(S \cup \{l,j\}))$, that is, the problem reduces to finding the subset of clients, say S', that includes j and l, and such that $\overline{x}(\delta(S'))$ is minimum. Build a graph G' by adding to $G[\overline{x}]$ a source node s linked to clients j and l with infinite weight edges, and a sink node t connected to all the depots, also with infinite weight edges. It is easy to see that S' can be obtained by computing the minimum weighted cut separating s and t in graph G' and then removing s from the set of nodes defining that cut on the s side. If $\overline{x}(\delta(S'))$ is less than $2A(j,l,I')$ the path constraint is violated for $S = S' \backslash \{l,j\}$, the pair of clients $\{l,j\}$ and the subset of depots I'; otherwise, no path constraint is violated for the given pair of clients $\{j,l\}$.

5.2.4. Separation of depot capacity and depot degree constraints

We have implemented three heuristic separation procedures for depot capacity constraints (11) and (12), and depot degree constraints (19) that use as starting point the same sets S generated by the separation algorithms for WCAP constraints explained above.

The procedure for separating (11) only considers sets $S \subseteq J$, such that $D(S) > \min\{W_i : i \in I\}$ and $\overline{x}(S : J \backslash S) < 0.9$ and, for these sets, we look for subsets $S' \subset S$ and $I' \subset I$ that define a violated constraint (11). For every $i \in I$ let $b_i = \overline{x}(S : \{i\})+2\overline{w}(S : \{i\})$; we begin with $I' = I$ and remove iteratively from I' the depot i for which the value of b_i is minimum among those remaining in I', until $D(S) > \sum_{i \in I'} W_i$. Then, the procedure looks for a subset $S' \subset S$: the sum $\overline{w}(j) = \sum_{i \in I \backslash I'} \overline{w}_{ij}$ is computed for all $j \in S$, and the nodes of S are added to S' one by one in non-increasing order of $\overline{w}(j)$, while $D(S \backslash S') > \sum_{i \in I'} W_i$ holds. Finally, the violation of (11) is checked.

The procedure for separating (12) is very similar. It is applied to the same sets $S \subseteq J$ as before. The subset $I' \subseteq I$ and depot $i \in I'$ defining the inequality are determined as follows. Initially I' contains all the depots l for which $b_l > 0$. Then a sequence of subsets I' is generated by removing each time the depot l with the lowest value of b_l among those remaining in I'. For each subset I' in the sequence and each depot $i \in I'$, the inequality (12) is checked for possible violation.

Finally, constraints (19) are simply checked for possible violation for the same sets $S \subseteq J$ as before, satisfying also $D(S) \le Q$, and for every depot $i \in I$.

5.2.5. Separation of co-circuit constraints

Two different procedures have been developed to separate co-circuit constraints (14). Both procedures are better understood if the constraints are rewritten in the following way:

$$\sum_{e \in \delta(S) \backslash F} x_e + \sum_{e \in F} (1-x_e) \ge 1 \quad \forall S \subseteq J \cup I,\ \forall F \subseteq \delta(S), |F| \text{ odd} \tag{25}$$

The first procedure is similar to the one used by Ghiani and Laporte [13] to separate co-circuit constraints in the Rural Postman Problem. Given a set $S \subseteq J$ it is easy to find the set $F \subseteq \delta(S)$ for which the LHS of (25) is minimized. Let $F=\{e\in\delta(S):\bar{x}_e\geq 0.5\}$, if $|F|$ is even, either one edge of F is removed or one edge in $\delta(S)\backslash F$ is added to F; the chosen edge is the one producing the smallest variation in the left hand side of (25). Finally, the violation of (14) is checked. This method is applied to the sets $S=\{v\}$, for each $v\in I\cup J$.

The second procedure is an exact algorithm which is the adaptation to our problem of the one proposed by Letchford et al. [21] to separate blossom inequalities. This algorithm computes the tree T of minimum weighted cuts of the graph induced by the edges $e\in E$ such that $\bar{x}_e>0$ with edge weights $h_e=\min\{x_e,\ 1-x_e\}$. Let S^f be the set of nodes that defines the minimum weighted cut associated with an edge f of the tree T, then the weight of this cut is equal to the left hand side of the corresponding constraint (25). Therefore, we can detect a violated co-circuit constraint (25), if any exists, by applying the greedy algorithm explained above to the sets S^f that correspond to edges of T with $h_e<1$.

5.2.6. Separation of CVRP constraints

Any LP solution of our problem can be transformed into an LP solution of a CVRP instance by shrinking all the depots into a single one. Then, any procedure for separating CVRP constraints can be applied to it. We have used the procedures provided by [22] to separate the following CVRP constraints: capacity, strengthened combs, homogeneous multistars, generalized large multistar and hypotours. As already mentioned the procedure that separates capacity constraints has been used in fact to find violated WCAP constraints. The interested reader can find a detailed description of these procedures in Lysgaard et al. [23].

5.3. Separation strategy

An important issue in a Branch-and-Cut algorithm is to decide which separation procedures will be called and when. We have spent some time testing different strategies and finally, we decided to use strategies similar to the ones used by Lysgaard et al. [23].

Our strategy for the root node is as follows. First we call the connected components heuristic for capacity and WCAP constraints separation. If at least one constraint is violated, we reoptimize the LP. Otherwise, we run the shrinking heuristic for WCAP constraints and the greedy heuristic for the co-circuit constraints with sets $S=\{v\}$ for every $v\in I\cup J$. If at least one constraint violated by more than 0.2 is found, we reoptimize. Otherwise, we call the tabu search procedure for capacity constraints, the capacity separation routine of [22] and the exact procedures for path elimination, w-subtour elimination and co-circuit constraints. If at least one constraint violated by more than 0.2 is found, we reoptimize. Otherwise, the remaining separation routines for CVRP constraints of [22] are sequentially called until one of them finds a violated constraint. Moreover, the order in which they are called is changed cyclically at each iteration.

The strategy for the other nodes is simpler and the separation takes less time. First we call the connected components heuristic, the shrinking heuristic and the greedy heuristic for co-circuit constraints. If at least one constraint is found violated by more than 0.2 or more than 50 violated WCAP constraints are detected, we reoptimize the LP. Otherwise we call the tabu search procedure, the capacity separation of Lysgaard and the exact procedures for path elimination, w-subtour elimination and co-circuit constraints. If at least one violation greater than 0.2 is obtained, we reoptimize; otherwise we run the other separation routines for CVRP constraints of [22]. Nevertheless, these last routines are called at most once at each node of the Branch-and-Cut tree different from the root.

5.4. Branching strategy

The cutting plane algorithm is implemented in C++ and embedded in a Branch-and-Cut procedure using the commercial solver CPLEX 9 through Concert Technology. The branching variable and the next node to separate are determined by CPLEX but the choice can be controlled by different parameter settings. After some experiments we decided to use the Strong Branching strategy [4] with highest priority to branch on the y-variables. The MIPEmphasis parameter was set to 3, which means that the strategy is oriented to improve the Best Bound that is the lowest lower bound of the live nodes and, therefore, a valid lower bound for the problem. The Strong Branching strategy only branches on variables and it is quite effective because it compares the possible candidates before deciding which one is finally chosen.

Nevertheless, another technique that has produced good results in the CVRP is to branch on cutsets [25]. In order to have the possibility of branching on cutsets, we have used the following trick. At the end of the cutting plane algorithm at the root node, the WCAP and depot capacity constraints added to the LP are converted into equality constraints, by adding to each one a slack variable declared as integer. These new variables may then be selected by CPLEX for branching, which is equivalent to branch on the corresponding cutsets. The violated constraints found by the separation algorithms at each node of the Branch-and-Cut are added to the LP as local cuts, which means that they affect only the current node and all its descendants in the tree.

6. Computational study

6.1. Instances

The Branch-and-Cut code written in C++ was run on a Intel Core 2 Quad Q6700 at 2.66 GHz with 2 GB of RAM under Windows XP, and applied to three sets of randomly generated Euclidean instances from literature.

The *first set* contains 12 LRP instances with a set of identical capacitated vehicles and a set of possible locations for capacitated depots. Under the name Benchmark 1, it may be downloaded at http://prodhonc.free.fr/Instances/instances_us.htm. This set was created to evaluate metaheuristics for the general LRP in [29,30,28]. The instance names include the number of customers $n\in\{20,50\}$, the number of potential depots $m=5$, the number of clusters $\beta\in\{1,2,3\}$, and a letter a or b corresponding respectively to a vehicle capacity $Q=70$ or 150. The two instances with 50 customers, 2*BIS* and 2*bBIS*, contain two strongly separated clusters. The traveling costs c_{ij} are the Euclidean distances, multiplied by 100 and rounded up to the nearest integer. The other data (demands, depot capacities and fixed costs) are also integer. They were randomly generated as follows:

- the demands are drawn uniformly in the interval [11,20];
- the depot capacities ensure to open at least 2 or 3 depots.
- each depot has a specific fixed cost and the sum of fixed costs has the same order of magnitude as the total routing cost.

The *second set* of 10 instances comes from Barreto's Ph.D. thesis [8] on clustering heuristics for the LRP, see Benchmark 3 at http://prodhonc.free.fr/Instances/instances_us.htm. It is composed of LRP instances already used in the literature or derived from classical CVRP instances. Their traveling costs are equal to the Euclidean distances, but this time not rounded. In a few instances, depots have an unlimited capacity. Vehicle capacities and demands are sometimes huge. The set-up cost is the same for any depot and represents around 10% of the total cost. We kept the original instance names, adding the number of customers (21–88) and/or the number of potential depots (5–10) when missing.

The *third set* comprises 12 instances recently designed by Akca et al. [1], with 30 or 40 customers and five depot locations (the size is indicated in the instance names). Node coordinates are generated using a uniform distribution in [0,100]. The travel costs are computed as follows: first Euclidean distances are multiplied by 100, rounded to the nearest integer and then divided by 100. Vehicles and depots have limited capacities but induce no set-up cost. Finally, in order to assure that the triangle inequality holds, each travel distance is set to the length of the corresponding shortest path. These instances were kindly provided by Akca et al.

6.2. Results for sets 1 and 2

For the instances selected in sets 1 and 2, Table 1 shows the results obtained by the cutting plane algorithm (node zero of the Branch-and-Cut) and also by a partial Branch-and-Cut algorithm in which the only variables that are required to be integer are the y-variables (those indicating which depots are opened). The horizontal line delimits the two sets of instances. The first two columns give the instance name and an upper bound (*UB*). *UB* is the best-known solution value obtained by the metaheuristics published in [29,30,28], using various parameters. The next three columns display the lower bound obtained by the cutting plane method (*LB*), the percentage deviation of this lower bound from the upper bound (Gap $=(UB-LB)\times 100/UB$) and the running time in seconds (Time). The last three columns provide the same indicators, but for the partial Branch-and-Cut.

Two instances in set 1, 20-5-1b and 20-5-2b, are optimally solved by the cutting plane algorithm. On average, the cutting plane provides a lower bound with a gap of 5.51% with respect to the upper bound. This gap decreases to 1.99%, on average if the partial Branch-and-Cut with y-integer is run, and four additional instances, Gaspelle3-29-5, Gaspelle5-32-5, Gaspelle6-36-5 and Min27-5, are optimally solved.

The complete Branch-and-Cut algorithm has been run on the same sets of instances with a time limit of 7200 s (2 h). The results listed in Table 2 provide for each instance the best lower bound obtained, the corresponding gap, the total running time, the time spent to solve the LPs (Time LP) and to separate inequalities (Time SEP), and the total number of nodes in the Branch-and-Cut tree. An asterisk on the lower bound indicates an optimal solution value. Some instances were not solved to optimality, either because the time limit was reached before the Branch-and-Cut was completed (instances with a CPU time larger than 7200 s), or because of a lack of memory.

Note that 14 instances out of 22 were optimally solved, including all with up to 40 customers and three with 50 customers (50-5-2bBIS, 50-5-3b, and Christ50-5). As far as we know, this is the first time that instances of this size are solved to optimality for the general form of LRP addressed in this paper. In particular, our algorithm found an optimal solution with cost 562.22 for instance Gaspelle4-32-5, thus improving the best-known upper bound of 571.71.

Table 3 shows the numbers of constraints detected by our separation algorithms and added to the LP during the complete Branch-and-Cut: WCAP constraints (15)–(17), co-circuit (14), path elimination (13), depot capacity (11) and (12), depot degree (19), and finally the CVRP strengthened combs, multistar and hypotours. On average, 73.6% of these constraints correspond to WCAP constraints.

It is also interesting to see if the new families of constraints that do not belong to the former integer formulation are effective to strengthen the linear relaxation. Therefore we have made a version of the cutting plane algorithm that uses only the separation procedures for the WCAP, improved path elimination and depot capacity constraints. This version, called *base*, includes all the constraints that allow to cut-off non-feasible integer solutions. Three additional versions of the code, called *base+CVRP*, *base+co-circuit*, *base+depot degree*, were built by adding to the *base* code the separation procedures for the CVRP constraints, co-circuit constraints and depot degree constraints, respectively.

Table 4 shows the results obtained by these four versions of the cutting plane on instance sets 1 and 2. The second column shows the gap corresponding to the percentage deviation of the lower

Table 1
Sets 1 and 2, results of cutting plane and partial Branch-and-Cut.

Instance name	UB	Cutting plane			B&C y integer		
		LB	Gap	Time	LB	Gap	Time
20-5-1a	54793	50885.02	7.13	0.34	53530.17	2.30	0.86
20-5-1b	39104	39104.00	0.00	0.17	39104.00	0.00	0.19
20-5-2a	48908	47180.87	3.53	0.25	47983.72	1.89	0.53
20-5-2b	37542	37542.00	0.00	0.06	37542.00	0.00	0.06
50-5-1a	90111	79647.73	11.61	12.38	85962.35	4.60	54.41
50-5-1b	63242	58208.60	7.96	3.97	60024.67	5.09	8.11
50-5-2a	88298	81692.41	7.48	8.14	85341.21	3.35	35.66
50-5-2b	67340	63847.31	5.19	2.45	64862.96	3.68	6.05
50-5-2BIS	84055	82401.94	1.97	7.19	82852.79	1.43	11.81
50-5-2bBIS	51822	51474.95	0.67	1.55	51506.85	0.61	1.81
50-5-3a	86203	76504.61	11.25	6.88	83080.45	3.62	31.61
50-5-3b	61830	56916.60	7.95	3.14	60304.09	2.47	10.28
Christ50-5	565.60	508.22	10.15	2.70	556.71	1.57	62.75
Christ75-10	861.62	756.95	12.15	37.66	798.80	7.29	400.03
Das88-8	355.78	335.04	5.83	21.33	348.34	2.09	73.22
Gaspelle-21-5	424.90	408.27	3.91	0.22	421.69	0.75	0.58
Gaspelle2-22-5	585.11	583.49	0.28	0.14	584.51	0.10	0.17
Gaspelle3-29-5	512.10	487.93	4.72	0.41	512.10	0.00	0.86
Gaspelle4-32-5	571.71	527.87	7.67	0.61	555.27	2.88	1.11
Gaspelle5-32-5	504.33	486.90	3.46	0.39	504.33	0.00	0.50
Gaspelle6-36-5	460.37	447.51	2.79	0.72	460.37	0.00	2.23
Min27-5	3062.02	2889.81	5.62	0.27	3062.02	0.00	0.70
Average			5.51	5.04		1.99	31.98

Table 2
Sets 1 and 2, results of the complete Branch-and-Cut algorithm.

Instance name	B&C (time limit 7200 s)					
	LB	Gap	Time	Time LP	Time SEP	Nodes
20-5-1a	54 793*	0.00	2.41	1.68	0.73	33
20-5-1b	39 104*	0.00	0.13	0.11	0.02	0
20-5-2a	48 908*	0.00	2.81	2.00	0.81	55
20-5-2b	37 542*	0.00	0.06	0.05	0.02	0
50-5-1a	88 071.60	2.26	7212.25	6207.87	1004.38	2555
50-5-1b	62 377.00	1.37	5557.95	4603.97	953.98	7840
50-5-2a	87 315.89	1.11	7208.06	6183.63	1024.43	3152
50-5-2b	66 377.50	1.43	6013.58	4361.95	1651.63	9449
50-5-2BIS	83 660.81	0.47	7207.05	5681.45	1525.60	8450
50-5-2bBIS	51 822*	0.00	9.16	3.57	5.59	40
50-5-3a	84 654.02	1.80	7206.95	5970.14	1236.82	2853
50-5-3b	61 830*	0.00	96.86	60.45	36.41	203
Christ50-5	565.60*	0.00	181.06	129.71	51.36	209
Christ75-10	810.85	5.89	3017.83	2260.12	757.71	359
Das88-8	351.52	1.20	3279.17	2521.49	757.69	1150
Gaspelle-21-5	424.90*	0.00	0.63	0.47	0.16	11
Gaspelle2-22-5	585.11*	0.00	0.17	0.11	0.06	2
Gaspelle3-29-5	512.10*	0.00	1.03	0.48	0.55	13
Gaspelle4-32-5	562.22*	0.00	3.45	2.48	0.97	21
Gaspelle5-32-5	504.33*	0.00	0.50	0.25	0.25	1
Gaspelle6-36-5	460.37*	0.00	2.05	1.03	1.01	4
Min27-5	3062.02*	0.00	0.67	0.39	0.28	5
Average		0.71	2136.54	1726.97	409.57	

Table 3
Number of constraints found by the complete Branch-and-Cut algorithm.

Instance name	WCAP	Co-circuit	Path	Depot cap.	Depot degree	Combs	Multi-star	Hypotours
20-5-1a	499	115	57	18	20	3	52	13
20-5-1b	30	15	0	0	2	0	0	0
20-5-2a	286	62	40	14	8	3	76	13
20-5-2b	14	4	0	0	1	0	0	0
50-5-1a	5872	200	21	24	105	46	540	58
50-5-1b	3833	1030	126	10	97	360	254	23
50-5-2a	5785	357	68	9	72	18	439	46
50-5-2b	3506	1587	130	43	30	639	95	38
50-5-2BIS	3785	260	152	15	26	66	505	40
50-5-2bBIS	815	341	48	21	20	67	112	26
50-5-3a	5354	99	17	13	73	96	76	11
50-5-3b	4149	1120	140	17	87	413	78	7
Christ50-5	2779	671	23	3	96	206	26	1
Christ75-10	5941	983	30	1	206	130	291	264
Das88-8	2474	1607	144	34	151	179	76	59
Gaspelle-21-5	159	46	15	6	6	3	13	12
Gaspelle2-22-5	79	13	0	1	17	1	0	0
Gaspelle3-29-5	165	50	0	1	33	4	0	0
Gaspelle4-32-5	443	54	0	3	46	15	19	1
Gaspelle5-32-5	123	35	0	4	48	8	0	2
Gaspelle6-36-5	543	46	0	3	37	1	66	15
Min27-5	203	41	1	6	24	2	0	5

Table 4
Comparative of the efficiency of each family of constraints.

Instance name	*Base*	*Base+CVRP*		*Base+co-circuit*		*Base+depot deg.*	
	Gap	Gap	Gap closed	Gap	Gap closed	Gap	Gap closed
20-5-1a	7.51	7.17	4.47	7.44	0.89	7.51	0.00
20-5-1b	0.00	0.00		0.00		0.00	
20-5-2a	4.26	3.78	11.26	4.14	2.81	3.74	12.24
20-5-2b	0.00	0.00		0.00		0.00	
50-5-1a	13.11	12.38	5.50	13.18	−0.59	12.28	6.30
50-5-1b	9.15	8.74	4.49	9.04	1.26	8.32	9.11
50-5-2a	8.17	7.84	4.14	8.22	−0.56	7.86	3.88
50-5-2b	5.50	5.35	2.59	5.27	4.20	5.45	0.78
50-5-2BIS	2.48	2.29	7.87	2.48	0.11	2.17	12.60
50-5-2bBIS	0.71	0.67	5.94	0.71	0.85	0.71	0.00
50-5-3a	11.63	11.50	1.17	11.27	3.16	11.38	2.14
50-5-3b	8.80	8.76	0.44	8.96	−1.86	8.07	8.25
Christ50-5	11.44	11.38	0.52	11.06	3.25	10.54	7.85
Christ75-10	13.28	12.73	4.14	13.15	0.97	12.48	6.01
Das88-8	6.24	5.94	4.80	6.17	1.05	6.26	−0.39
Gaspelle-21-5	4.27	4.11	3.72	4.21	1.60	4.19	2.05
Gaspelle2-22-5	0.67	0.49	26.88	0.65	4.13	0.42	36.89
Gaspelle3-29-5	7.52	7.52	0.00	6.94	7.67	4.88	35.03
Gaspelle4-32-5	6.61	6.60	0.08	6.61	0.04	6.29	4.90
Gaspelle5-32-5	4.66	4.59	1.48	4.50	3.38	3.59	22.91
Gaspelle6-36-5	2.79	2.79	0.00	2.79	0.00	2.79	0.00
Min27-5	6.69	6.68	0.11	6.68	0.14	5.66	15.33
Average	6.16	5.97	4.48	6.07	1.62	5.66	9.29

bound obtained with the *base* code from the upper bound. For the other versions of the code, Table 4 shows the gap corresponding to the lower bound obtained with each version and also the closed gap. This closed gap is computed as $100(LB\,1-LB\,0)/(UB-LB\,0)$, where $LB\,0$ is the lower bound obtained with the *base* code, $LB\,1$ is the lower bound obtained after adding the corresponding family of constraints and UB is the best-known upper bound. On average depot degree constraints produce the largest closed gap (9.29%), but all the other types of constraints are also useful. However, it happens that in some instances the improvement is negligible.

The version with co-circuit constraints is sometimes worse than the *base* code, like in instance 50-5-3b. This behavior can be explained by the use of heuristic separation procedures for some constraints. It is possible that some constraints are violated in the LP solution but not detected by these procedures. Hence, the final set of constraints added to the LP may depend on the particular sequence of LP solutions that is generated during the cutting plane algorithm.

6.3. *Results for set 3*

The results for the instances from Akca et al. are listed in Tables 5 and 6. Table 5 has the same format as Table 1 but *UB* is now the solution value computed by Akca's heuristic algorithm and rounded up to one decimal. Table 6 is similar to Table 2, with

Table 5
Set 3, results of cutting plane and partial Branch-and-Cut.

Instance name	UB	Cutting plane			B&C *y* integer		
		LB	Gap	Time	LB	Gap	Time
cr30x5a1	819.6	784.53	4.28	0.70	788.20	3.83	0.89
cr30x5a2	823.5	770.68	6.41	0.53	791.44	3.89	1.17
cr30x5a3	702.3	694.67	1.09	0.52	702.29	0.00	0.72
cr30x5b1	880.1	813.36	7.58	0.47	858.95	2.40	1.58
cr30x5b2	825.4	789.27	4.38	0.50	822.34	0.37	0.88
cr30x5b3	884.7	856.93	3.14	0.95	871.44	1.50	1.94
cr40x5a1	928.2	841.71	9.32	1.14	901.68	2.86	12.59
cr40x5a2	888.4	809.65	8.86	0.94	867.30	2.38	7.48
cr40x5a3	947.3	874.76	7.66	2.34	925.07	2.35	12.56
cr40x5b1	1052.1	940.61	10.60	1.31	1001.88	4.77	7.31
cr40x5b2	981.6	894.07	8.92	1.38	967.49	1.44	5.83
cr40x5b3	964.4	914.11	5.21	1.48	950.62	1.43	4.42
Average			6.45	1.02		2.27	4.78

Table 6
Set 3, results of the complete Branch-and-Cut algorithm.

Instance name	Akca et al.		B&C (time limit 7200 s.)					
	Gap	Time	LB	Gap	Time	Time LP	Time SEP	Nodes
cr30x5a1	0.00	993.3	819.60*	0.00	50.22	40.11	10.11	775
cr30x5a2	2.55	10806.50	823.50*	0.00	53.89	37.86	16.03	799
cr30x5a3	0.00	917.90	702.30*	0.00	0.73	0.38	0.36	2
cr30x5b1	0.00	6420.60	880.10*	0.00	8.48	5.72	2.77	74
cr30x5b2	0.00	33.20	825.40*	0.00	1.09	0.55	0.55	7
cr30x5b3	0.00	41.70	884.70*	0.00	5.63	3.99	1.64	26
cr40x5a1	1.49	10882.80	928.20*	0.00	305.25	238.14	67.11	810
cr40x5a2	0.93	11052.90	888.40*	0.00	98.34	70.65	27.69	377
cr40x5a3	0.18	10862.00	947.30*	0.00	158.27	117.30	40.96	406
cr40x5b1	0.00	8084.60	1052.07*	0.00	3694.45	2436.83	1257.62	15399
cr40x5b2	0.00	862.50	981.60*	0.00	10.25	4.25	6.00	34
cr40x5b3	0.00	963.00	964.40*	0.00	11.36	5.60	5.76	33
Average	0.43	5160.1		0.00	366.50	246.78	119.72	

two new columns showing the results reported by Akca et al. [1] with a Branch-and-Price algorithm running on a 1.8 GHz PC with 2 GB of RAM under Linux, with a time limit of 10 800 s (3 h). Our algorithm is able to solve all these instances to optimality, while the Branch-and-Price algorithm of Akca et al. fails to solve four instances in spite of its larger time limit. For the eight instances solved by both methods, Akca et al. needed 1518.4 s on average, while our method required 472.78 s only, a smaller duration even if we consider the different processor clock speeds (1.8 GHz vs. 2.66 GHz) of the computers used.

7. Conclusions

This paper presents the first Branch-and-Cut algorithm for solving the Location-Routing Problem with capacitated depots and vehicles. It is based on an integer formulation using only binary variables. The formulation is strengthened by new families of constraints and by valid inequalities derived from the CVRP. Two new exact polynomial algorithms are developed to separate path elimination and w-subtour elimination constraints, and a number of heuristic separation procedures have been designed for all the families of constraints.

The method is tested on three sets of instances from literature, comprising 34 instances with 20–88 customers and 5–10 potential depot locations. The computational testing shows that 26 instances with five potential depots are solved to optimality, including all instances up to 40 customers and three instances with 50 customers. These results are encouraging since the LRP has a much larger solution space than the CVRP. However, further research is still required to develop an exact method able to cope with larger real-life instances.

Location-Routing Problems may occur in short term planning horizon, in emergency situations to supply a devastated region for instance, but most applications are raised by the design of distribution networks, with long planning horizons. In that case, the LRP is pertinent only if the set of customers and their demands are relatively stable over the planning horizon considered. A promising research direction can be the study of situations with varying demands over several weeks or months. This could be adequately captured by a *Periodic Location-Routing Problem* (PLRP). Other possible perspectives include all the additional constraints addressed in rich vehicle routing problems (e.g., time windows), and Location-Routing Problems with two echelons, in which the open depots must be supplied from a main depot.

Acknowledgements

The contribution by J.M. Belenguer and E. Benavent has been partially supported by the Ministerio de Ciencia y Tecnología of Spain through projects DPI2005-04796 and MTM2006-14961-C05-02.

References

[1] Akca Z, Berger RT, Ralphs TK. A branch-and-price algorithm for combined location and routing problems under capacity restrictions. In: Chinneck JW, Kristjansson B, Saltzman MJ, editors. Operations research and cyber-infrastructure, vol. 47. US: Springer; 2009. p. 309–30.

[2] Albareda-Sambola M, Díaz JA, Fernández E. A compact model and tight bounds for a combined location-routing problem. Computers & Operations Research 2005;32:407–28.

[3] Albareda-Sambola M, Fernández E, Laporte G. Heuristic and lower bounds for a stochastic location routing problem. European Journal of Operational Research 2007;179:940–55.

[4] Applegate D, Bixby RE, Chvàtal V, Cook W. Finding cuts in the TSP. Technical Report 95-05, DIMACS; 1995.

[5] Augerat P, Belenguer JM, Benavent E, Corberán A, Naddef D. Separating capacity inequalities in the CVRP using tabu search. European Journal of Operational Research 1998;106:546–57.

[6] Augerat P, Belenguer JM, Benavent E, Corberán A, Naddef D, Rinaldi G. Computational results with a branch-and-cut code for the capacitated vehicle routing problem. Technical Report RR949-M, ARTEMIS-IMAG, France; 1995.

[7] Barahona F, Grötschel M. On the cycle polytope of a binary matroid. Journal of Combinatorial Theory 1986;40:40–62.

[8] Barreto SS. Análise e Modelização de Problemas de localização-distribuição [Analysis and modelling of location-routing problems]. PhD thesis, University of Aveiro, campus universitário de Santiago, 3810-193 Aveiro, Portugal; October 2004 [in Portuguese].

[9] Belenguer JM, Benavent E, Prins C, Prodhon C, Wolfler-Calvo R. A branch and cut method for the capacitated location-routing problem. In: Proceedings of the IEEE international conference on service systems and service management (ICSSSM'06), vol. 2; 2006. p. 1541–6.

[10] Bruns A, Klose A. A "locate first-route second" heuristic for a combined location-routing problem. In: Zimmermann U, Derigs U, Gaul W, Möhring RH, Schuster KP, editors. Operations research proceedings 1996. Springer; 1996. p. 49–54.

[11] Chan Y, Carter WB, Burnes MD. A multiple-depot, multiple-vehicle, location-routing problem with stochastically processed demands. Computers & Operations Research 2001;28(8):803–26.

[12] Chien TW. Heuristic procedures for practical-sized uncapacitated location-capacitated routing problems. Decision Science 1993;24(5):995–1021.

[13] Ghiani G, Laporte G. A branch-and-cut algorithm for the undirected rural postman problem. Mathematical Programming 2000;87(3):467–81.

[14] Ghiani G, Laporte G. Location-arc routing problems. OPSEARCH 2001;38(2): 151–9.

[15] Golden B, Raghavan S, Wasil E, editors. The vehicle routing problem: latest advances and new challenges. In: Operations research/computer science interfaces series, vol. 43. Springer; 2008.

[16] Labadi N. Problèmes tactiques et stratégiques en tournées sur arcs [Tactical and strategic problems in arc routing]. PhD thesis, Université de Technologie de Troyes, Troyes, France; 2003 [in French].

[17] Laporte G, Louveaux F, Mercure H. Models and exact solutions for a class of stochastic location-routing problems. European Journal of Operational Research 1989;39:71–8.
[18] Laporte G, Nobert Y. An exact algorithm for minimizing routing and operating costs in depot location. European Journal of Operational Research 1981;6: 224–6.
[19] Laporte G, Nobert Y, Arpin D. An exact algorithm for solving a capacitated location-routing problem. Annals of Operations Research 1986;6:293–310.
[20] Laporte G, Nobert Y, Taillefer S. Solving a family of multi-depot vehicle routing and location-routing problems. Transportation Science 1988;22(3): 161–72.
[21] Letchford AN, Reinelt G, Theis DO. Odd minimum cut sets and b-matchings revisited. SIAM Journal on Discrete Mathematics 2008;22(4):1480–7.
[22] Lysgaard J. A CVRPSEP package, 〈http://www.hha.dk/~lys/CVRPSEP.htm〉; 2004.
[23] Lysgaard J, Letchford AN, Eglese RW. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming 2004;100(2):423–45.
[24] Min H, Jayaraman V, Srivastava R. Combined location-routing problems: a synthesis and future research directions. European Journal of Operational Research 1998;108(1):1–15.
[25] Naddef D, Rinaldi G. Branch-and-cut algorithms for the capacitated VRP. In: Toth P, Vigo D, editors. The vehicle routing problem, SIAM monographs on discrete mathematics applications, vol. 9; 2002. p. 53–84.
[26] Nagy G, Salhi S. The many-to-many location-routing problem. Top 1998;6(2): 261–75.
[27] Padberg MW, Rinaldi G. A branch and cut algorithm for the resolution of large-scale symmetric traveling salesmen problems. SIAM Review 1991;33:60–100.
[28] Prins C, Prodhon C, Ruiz A, Soriano P, Wolfler-Calvo R. Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic. Transportation Science 2007;41(4):470–83.
[29] Prins C, Prodhon C, Wolfler-Calvo R. A memetic algorithm with population management ($MA \mid PM$) for the capacitated location-routing problem. In: Gottlieb J, Raidl GR, editors. Proceedings of EvoCOP2006 (evolutionary computation in combinatorial optimization: 6th european conference, Budapest, Hungary, April 10–12, 2006), Lecture notes in computer science, vol. 3906. Springer; 2006. p. 183–94.
[30] Prins C, Prodhon C, Wolfler-Calvo R. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR—A Quarterly Journal of Operations Research 2006;4:221–38.
[31] Ralphs TK, Kopman L, Pulleyblank WR, Trotter LE. On the capacitated vehicle routing problem. Mathematical Programming 2003;94:343–59.
[32] Salhi S, Rand GK. The effect of ignoring routes when locating depots. European Journal of Operational Research 1989;39(2):150–6.
[33] Srivastava R. Alternate solution procedures for the locating-routing problem. OMEGA International Journal of Management Science 1993;21(4):497–506.
[34] Toth P, Vigo D, editors. The vehicle routing problem. SIAM monographs on discrete mathematics Applications, vol. 9, 2002.
[35] Tuzun D, Burke LI. A two-phase tabu search approach to the location routing problem. European Journal of Operational Research 1999;116(1):87–99.
[36] Wu TH, Low C, W Bai J. Heuristic solutions to multi-depot location-routing problems. Computers & Operations Research 2002;29(10):1393–415.

.

Computers & Operations Research 38 (2011) 1319–1334

Contents lists available at ScienceDirect

Computers & Operations Research

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

A GRASP with evolutionary path relinking for the truck and trailer routing problem

Juan G. Villegas [a,b,c,*], Christian Prins [a], Caroline Prodhon [a], Andrés L. Medaglia [b], Nubia Velasco [b]

[a] Laboratoire d'Optimisation des Systèmes Industriels (LOSI), Institut Charles Delaunay, Université de Technologie de Troyes, BP 2060, 10010 Troyes Cedex, France
[b] Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes, A.A. 4976, Bogotá D.C., Colombia
[c] Departamento de Ingeniería Industrial, Universidad de Antioquia, Calle 67 No. 53 - 108, A.A. 1226, Medellin, Colombia

ARTICLE INFO

Available online 30 November 2010

Keywords:
Truck and trailer routing problem (TTRP)
Greedy randomized adaptive search procedures (GRASP)
Variable neighborhood search (VNS)
Path relinking
Vehicle routing problem

ABSTRACT

In the truck and trailer routing problem (TTRP) a heterogeneous fleet composed of trucks and trailers has to serve a set of customers, some only accessible by truck and others accessible with a truck pulling a trailer. This problem is solved using a route-first, cluster-second procedure embedded within a hybrid metaheuristic based on a greedy randomized adaptive search procedure (GRASP), a variable neighborhood search (VNS) and a path relinking (PR). We test PR as a post-optimization procedure, as an intensification mechanism, and within evolutionary path relinking (EvPR). Numerical experiments show that all the variants of the proposed GRASP with path relinking outperform all previously published methods. Remarkably, GRASP with EvPR obtains average gaps to best-known solutions of less than 1% and provides several new best solutions.

1. Introduction

In the truck and trailer routing problem [7] a heterogeneous fleet composed of m_t trucks and m_r trailers ($m_r < m_t$) serves a set of customers $N = \{1,\ldots,n\}$ from a main depot, denoted by 0. Each customer $i \in N$ has a non-negative demand q_i; the capacities of the trucks and the trailers are Q_t and Q_r, respectively; and the distance c_{ij} between any two points $i,j \in N \cup \{0\}$ $(i \neq j)$ is known. Some customers with limited maneuvering space or accessible through narrow roads must be served only by a truck, while other customers can be served either by a truck or by a *complete vehicle* (i.e., a truck pulling a trailer). These incompatibility constraints create a partition of N into two subsets: the subset of *truck customers* N_t accessible only by truck; and the subset of *vehicle customers* N_v accessible either by truck or by a complete vehicle. The objective of the TTRP is to find a set of routes of minimum total distance such that: each customer is visited in a route performed by a compatible vehicle; the total demand of the customers visited in a route does not exceed the capacity of the allocated vehicle; and the number of required trucks and trailers is not greater than m_t and m_r, respectively. Being an extension of the well-known vehicle routing problem (VRP), the TTRP is NP-Hard. For updated reviews of the VRP and its extensions the reader is referred to the books by Toth and Vigo [53] and Golden et al. [24], and the introductory tutorial by Laporte [30].

A solution of the TTRP may have three types of routes: *pure truck routes* performed by a truck visiting customers in N_v and N_t; *pure vehicle routes* performed by a complete vehicle serving only customers in N_v; and finally *vehicle routes with subtours*. The latter are composed of a main tour performed by the complete vehicle visiting only customers in N_v, and one or more subtours, in which the trailer is detached at a vehicle customer location, to visit (with the truck) one or more customers in N_t and probably some customers in N_v. The parking place of the trailer is called the root of the subtour.

Fig. 1 depicts a solution of the TTRP with $m_t = 4, m_r = 3, Q_t = 6$, and $Q_r = 10$. Solid lines represent segments traversed by a complete vehicle and dashed lines represent segments traversed by a truck. Fig. 1 illustrates some of the special features of the TTRP: (i) only the vehicle customers visited in the main tour can be used as roots in vehicle routes with subtours; (ii) the total demand of the customers of a subtour must not exceed Q_t; (iii) the total demand of the customers of all subtours visited on a vehicle route with subtours may exceed Q_t (but not Q_t+Q_r), because at the root of a subtour it is possible to transfer goods between the truck and the trailer; (iv) several subtours may have the same root; (v) the first customer of a vehicle route with subtours cannot be a truck customer, because in that case the trailer would have been detached at the main depot due to the accessibility constraint of the first customer, giving rise to a pure truck route.

* Corresponding author at: Departamento de Ingeniería Industrial, Universidad de Antioquia, Calle 67 No. 53 - 108, A.A. 1226, Medellín, Colombia. Tel: +574 2195575.
E-mail addresses: jvillega@udea.edu.co (J.G. Villegas), christian.prins@utt.fr (C. Prins), caroline.prodhon@utt.fr (C. Prodhon), amedagli@uniandes.edu.co (A.L. Medaglia), nvelasco@uniandes.edu.co (N. Velasco).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2010.11.011

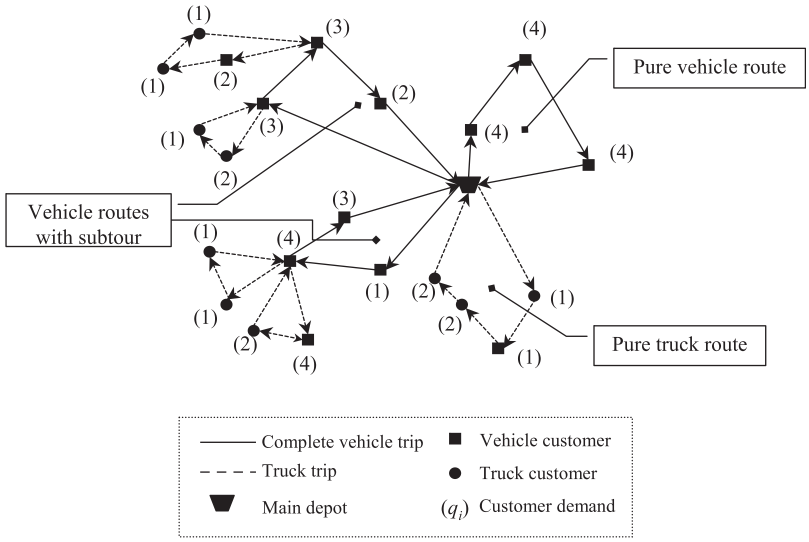

Fig. 1. Feasible solution for the TTRP.

For the solution of the TTRP we present a hybrid metaheuristic based on a greedy randomized adaptive search procedure (GRASP), a variable neighborhood search (VNS) and a path relinking (PR). The remainder of this paper is organized as follows. Section 2 motivates the TTRP with some practical applications and presents the relevant literature on the TTRP and other related problems. Section 3 describes the hybrid metaheuristic and its components. Section 4 presents a computational evaluation of different variants of the hybrid metaheuristic on a set of publicly available test problems and their comparison against other methods from the literature. Finally, Section 5 presents some conclusions. Appendix A summarizes the notation used throughout the paper.

2. Literature review

Practical applications of the TTRP appear mainly in collection and delivery operations in rural areas or crowded cities with accessibility constraints. Semet and Taillard [48] used a tabu search to solve a TTRP with time windows, site dependencies and heterogeneous fleet arising in the distribution operations of a chain of grocery stores in Switzerland. Gerdessen [21] described two possible applications of the TTRP. The first one arises in the distribution of dairy products in the Netherlands, where the use of trucks with trailers is common. However, customers located in crowded cities cannot be served by the complete vehicle, thus the trailers must be left in parking lots before reaching these customers. The second application is related with the distribution of compound animal feed in rural regions, customers reachable through narrow roads or bridges must be served by the truck after leaving the trailer parked in a proper place.

Milk collection is another known practical application of the TTRP. Hoff and Løkketangen [28] presented a tabu search algorithm for the solution of a routing problem for milk collection in Norway. They modeled the milk collection using a multi-depot TTRP variant, where N_v is empty and the parking places for the trailers are not associated with customer locations. Likewise, Caramia and Guerriero [6] used the Heterogeneous Milk Collection with Heterogeneous Fleet (HMCHF) problem to model and optimize the milk collection of an Italian dairy company. The HMCHF problem can be seen as a TTRP variant with multi-compartments, route-length constraints, and heterogeneous trucks and trailers.

Chao [7] introduced the TTRP and also proposed a tabu search metaheuristic based on a cluster-first, route-second approach. The clustering phase solves a relaxed generalized assignment problem (RGAP) to allocate customers to routes. The RGAP is solved by rounding the solution of its linear programming relaxation; this rounding may produce infeasible solutions with overcapacity utilization. The second phase uses a cheapest insertion heuristic to sequence the customers within each route. The insertion heuristic treats pure vehicle routes and pure truck routes as classical traveling salesman problems (TSPs); on the other hand, when constructing vehicle routes with subtours the insertion procedure takes into account the accessibility constraints. In a third step a multiple-neighborhood improvement procedure with a penalized objective function is used to repair infeasible solutions and improve feasible ones. The neighborhoods are reallocations and exchanges of customers between routes, a specialized neighborhood that changes the roots of the subtours, and a 2-opt [18] refinement for every route and subtour. The fourth and final phase is a hybrid tabu search/deterministic annealing method that reuses some of the neighborhoods of the previous phase, and implements a tabu restriction that forbids moves that increase the objective function over a certain threshold. The tabu search has one diversification stage and one intensification stage, executed in sequence. The search is restarted several times from the best solution found so far.

In the same vein, Scheuerer [45] proposed two constructive methods and a tabu search for the TTRP. The first constructive heuristic, called T-Cluster, is a cluster-based insertion heuristic that constructs routes sequentially. The insertion of each customer into a route is followed by a steepest descent improvement procedure with three neighborhoods: a root refining approach (for vehicle routes with subtours [7]), 2-opt [18] and Or-opt [33]. The second constructive heuristic, called T-Sweep is an adaptation for the TTRP of the sweep heuristic of Gillett and Miller [22], followed by the same steepest descent procedure. In both methods, it is possible to produce infeasible solutions, because capacity violation of the last route is allowed when there are unrouted customers and no more vehicles available. Starting from the solution obtained with any of the above constructive heuristics, the tabu search method explores the neighborhood generated using reallocations and exchanges of subsets of customers between routes and subtours, and the procedure that changes the roots of vehicle routes with subtours [7]. After the acceptance of a solution, 2-opt and Or-opt procedures improve each modified tour. The search explores infeasible solutions using a penalized objective function and strategic oscillation following the approach of Cordeau et al. [9].

Neighborhood reduction strategies are used to speed-up the evaluation of moves. The search is restarted using the best solution found so far as an intensification mechanism.

Lin et al. [32] developed a very effective simulated annealing (SA) for the TTRP. Their SA uses an indirect representation of the solutions using a permutation of the customers with additional dummy zeros to separate routes and terminate subtours, along with a vector of binary variables of length $|N_v|$, representing the type of vehicle used to serve each vehicle customer (0 for a complete vehicle, and 1 for a truck). A specialized procedure decodes the permutation into a TTRP solution using the information of the binary vector. Since the decoding procedure may fail to find feasible solutions with respect to the availability of trucks and trailers, a route combination approach is used to reduce the number of required trucks and trailers, and within the simulated annealing heuristic a penalty term is added to the objective function to guide the search towards feasible regions. To solve the problem, the authors use a rather standard simulated annealing procedure with three neighborhoods applied to the indirect representation. For the permutation the neighborhoods are: reinsertion of a randomly selected customer or exchange of the position of a random pair of customers, whereas for the binary vector the neighborhood is defined by flipping the type of vehicle serving a randomly selected vehicle customer. To increase the chance of obtaining high quality solutions, half of the time the move performed is the best of several random trials of the selected neighborhood. By relaxing the truck and trailer availability constraints Lin et al. [31] proposed the relaxed TTRP (RTTRP). Reusing their SA, these authors discovered a non-trivial trade-off between the fleet size and the total distance.

Caramia and Guerriero [5] designed a mathematical-programming based heuristic that also employs the cluster-first, route-second approach. Their method solves two subproblems sequentially. The first one, called customer-route assignment problem (CAP) assigns the customers to valid routes seeking to reduce the size of the fleet. Then, given the assignment of customers to routes, the route-definition problem (RDP) minimizes the tour length of each route using a TSP like model without subtour elimination constraints. Since the RDP may produce solutions with disconnected subtours, an edge-insertion heuristic is used to properly connect subtours to the main route. For pure truck routes and pure vehicle routes the heuristic builds a single tour; while for vehicle routes with subtours, the heuristic selects the root of each subtour and constructs the main tour. The authors embedded these two models within an iterative mechanism that adds new constraints to the CAP based on the information of the RDP solution. This restarting mechanism is intended to diversify the search, and includes a tabu search mechanism that forbids (in the CAP) customers route assignments already explored in previous iterations of the algorithm.

In the literature several researchers have studied other vehicle routing problems with trailers that deviate from the TTRP. Semet [47] presented the partial accessibility constrained vehicle routing problem (PACVRP), in which each parking place for the trailer is restricted to have only one departing subtour. He provided an integer programming formulation and developed a cluster-first, route-second approach for the PACVRP. His article mainly discusses the clustering phase modeled with an extended generalized assignment problem, which is solved using Lagrangian relaxation embedded within branch and bound. Gerdessen [21] tackled the vehicle routing problem with trailers (VRPT) using constructive and local search heuristics. The VRPT differs from the TTRP in several simplifying assumptions: in the VRPT there are no accessibility constraints, instead a different service time is incurred if a customer is visited by a complete vehicle or by a truck; all the customers have unit demand, and each trailer is parked exactly once.

Drexl [13] proposed the vehicle routing problem with trailer and transshipments VRPTT. In the VRPTT the customers have time windows; the routing cost is vehicle dependent (i.e., if one arc is traversed by a complete vehicle it has a different cost than if it is traversed just by a truck); parking places (transshipment locations) differ from customer locations; and the assumption of a fixed truck-trailer assignment is dropped, so that a trailer may be pulled by any compatible truck in different routes. The author used a branch-and-cut method to solve the VRPTT and showed that only very small instances of this problem can be solved to optimality.

Recently, Villegas et al. [54] have studied the single truck and trailer routing problem with satellite depots (STTRPSD) in which a truck with a detachable trailer based at a main depot has to serve the demand of a set of customers accessible only by truck. Therefore, before serving the customers, the trailer is detached in appropriated parking places (called trailer points of satellite depots) where goods are transferred between the truck and the trailer. To solve the STTRPSD they proposed a multi-start evolutionary local search and a hybrid metaheuristic based on GRASP and variable neighborhood descent (VND). In their computational experiments, on a set of randomly generated instances, multi-start evolutionary local search outperformed GRASP/VND in terms of solution quality and running time.

3. GRASP/VNS with path relinking

GRASP is a memory-less multi-start metaheuristic in which a local search is applied to initial solutions obtained with a greedy randomized heuristic [17]. Even though GRASP has not been widely used for the solution of vehicle routing problems [20], GRASP-based hybrid metaheuristics have achieved competitive results in different routing problems such as the VRP [36], the capacitated location-routing problem [14] and the capacitated arc-routing problem with time windows [39].

Resende and Ribeiro [43] reported that the performance of GRASP can be enhanced by using reactive fine tuning mechanisms, multiple neighborhoods, and path relinking. Along this line, our hybrid metaheuristic includes VNS as the local search component and uses PR in different strategies. A description of the components and the general structure of the hybrid GRASP/VNS with path relinking follows.

3.1. Greedy randomized construction

Contrary to most of the solution methods for TTRP-related problems [5–7,28,45,47,48] that use a natural cluster-first, route-second approach, in this paper we use a route-first, cluster-second (RFCS) procedure for the randomized construction of GRASP. Even though in the 80s Beasley [1] introduced route-first, cluster-second heuristics for the VRP, it was only twenty years later that Prins [35] unveiled its potential as a component of metaheuristics for routing problems. The fundamental idea is to take a giant tour $T=(0,t_1,\dots,t_i,\dots,t_n,0)$ visiting all the customers and break it into VRP feasible routes using a tour splitting procedure. We follow the same spirit for the TTRP.

The randomized route-first, cluster-second heuristic follows three steps. First, a randomized nearest neighbor heuristic with a *restricted candidate list* (RCL) of size κ constructs a giant tour $T=(0,t_1,\dots,t_i,\dots,t_n,0)$, where t_i represents the customer in the i-th position of the tour. Note that T visits all the customers in N, ignoring the capacity of the vehicles and the accessibility constraints of customers in N_t. Second, we define an auxiliary acyclic graph $H=(X,U,W)$ where the set of nodes X contains a dummy node 0 and n nodes numbered 1 through n, where node i represents customer t_i (i.e., the customer in the i-th position of T); the arc set U contains one arc $(i-1,j)$ if and only if the subsequence $(t_i,\dots,t_j)$ can be served in a feasible route; and the weight $w_{i-1,j}$ of arc $(i-1,j)$ is the total distance of the corresponding route. Third, the shortest path between nodes 0 and n in H represents a TTRP solution S, where the cost of the shortest path corresponds to the total distance of S and the arcs in the shortest path represent its routes.

Note that to adapt the route-first, cluster-second (RFCS) approach for the TTRP it is necessary to take into account its complicating elements, namely, the accessibility constraints and the heterogeneous fixed fleet. We manage the accessibility constraints when building H and take into account the heterogeneous fixed fleet while solving the shortest path on H.

The arc set U in H has three types of arcs, each one representing a type of route. Before adding arc $(i-1,j)$ to U we perform a feasibility test for route $R_{ij}=(0,t_i,\ldots,t_j,0)$. Let $Q_{ij}=\sum_{u=i}^{j} q_{t_u}$ be the total demand of R_{ij}. If $Q_{ij}<Q_t$ then R_{ij} is feasible (regardless of the type of customers assigned to it), and it is a pure truck route. On the other hand, if $Q_t<Q_{ij}\le Q_t+Q_r$, R_{ij} is feasible if t_i is a vehicle customer, and the type of route represented by arc $(i-1,j)$, depends on the customers assigned to R_{ij}. If all customers belong to N_v then R_{ij} is a pure vehicle route (without subtours). But, if there is at least one customer in N_t, then R_{ij} is a vehicle route with subtours. Finally, if $Q_{ij}>Q_t+Q_r$ the route is infeasible and the arc dropped from set U.

Since the triangle inequality holds, the cost of pure truck routes and pure vehicle routes is easily calculated with $c(R_{ij})=c_{0t_i}+\sum_{u=i}^{j-1} c_{t_u,t_{u+1}}+c_{t_j0}$, because its structure corresponds to a single tour. On the contrary, the total distance of vehicle routes with subtours is calculated using an optimization subproblem that selects the parking places for the trailer, and builds the main tour and subtours. This subproblem can be seen as a restricted version of the single truck and trailer routing problem with satellite depots (STTRPSD) [54], in which the satellite depots correspond to the vehicle customers included in R_{ij}.

To solve the restricted STTRPSD associated with the vehicle route with subtours R_{ij} we use a dynamic programming method in which state $[l,m]$ $(i\le l\le j, l: t_l\in N_v; l\le m\le j)$ represents the use of vehicle customer t_l as the root of a subtour ending at customer t_m. The initial state $[i,i]$ represents the departure of the complete vehicle from the main depot to the first vehicle customer. By definition we include states $[l,l]\forall l: t_l\in N_v$ to represent the movement of the complete vehicle from the root of a subtour to the next vehicle customer performing an empty subtour of null cost.

Let F_{lm} be the cost of state $[l,m]$, and θ_{lkm} be the cost of a subtour $ST=(t_l,t_k,\ldots,t_m,t_l)$ rooted at vehicle customer t_l and visiting customer t_k to t_m. In order to find the parking places for the trailer in route R_{ij} we use a recurrence relation with three cases. The first case represents the initial state ($l=i$ and $m=i$) and corresponds to the trip of the complete vehicle from the depot to the first vehicle customer t_i. Its cost is given by $F_{lm}=c_{0t_i}$. Then, the second case ($l=i$ and $i<m\le j$) analyses the possibility of performing all the subtours based at vehicle customer t_i. The cost of these states is given by $F_{lm}=\min_k\{F_{ik}+\theta_{i,k+1,m} \mid k<m: \sum_{u=k+1}^{m} q_{t_u}\le Q_t\}$.

Finally, the general case ($i<l\le j, t_l\in N_v; l\le m\le j$) has two terms. The first term includes the alternative of having several subtours rooted at vehicle customer t_l, while the second term includes the movement of the complete vehicle from vehicle customer t_h to perform a subtour rooted at vehicle customer t_l. The cost of these states is given by:

$$F_{lm}=\min\left\{\min_{k}\left\{F_{lk}+\theta_{l,k+1,m} \mid k<m: \sum_{u=k+1}^{m} q_{t_u}\le Q_t\right\},\right.$$
$$\left.\min_{h,k}\left\{F_{hk}+c_{t_h,t_l}+\theta_{l,k+2,m} \mid h<l: t_h\in N_v; k=l-1: \sum_{u=k+2}^{m} q_{t_u}\le Q_t\right\}\right\}$$

In all cases, the additional conditions over k check the capacity constraints of the truck while performing the subtours; and the conditions over h assure that the main tour only visits vehicle customers. The recursive equation that includes the three cases is presented in Appendix B.

Since states $[l,m]$ do not include the return from the last vehicle customer to the main depot, the cost of the route is finally calculated as $c(R_{ij})=\min_{i\le l\le j: t_l\in N_v}\{F_{lj}+c_{t_l,0}\}$. To solve the STTRPSD we use a similar approach to that used by Villegas et al. [54]. See Appendix B for additional details of the solution procedure.

Once we have generated the auxiliary graph H, we solve a shortest path problem to find the optimal partition of T into a TTRP feasible solution. The limited fleet is taken into account at this stage, thus we solve a resource-constrained shortest path problem (RCSPP) in H, where the resources are the available trucks and trailers. Each arc $(i-1,j)$ in U has three attributes: the distance of

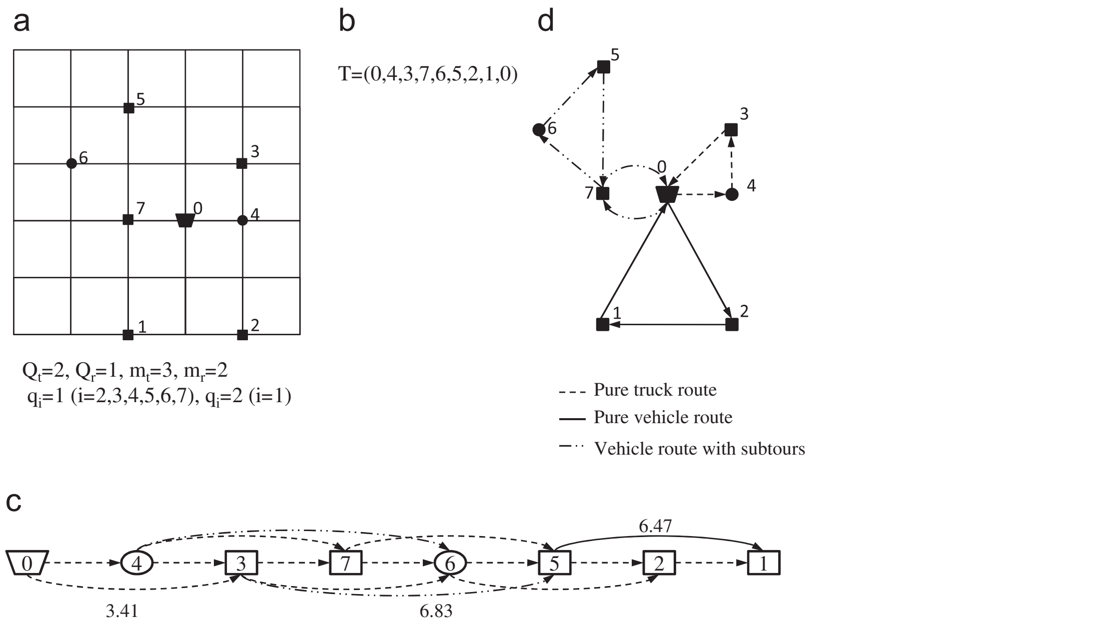

Fig. 2. Example of the route-first cluster-second procedure for the TTRP used in the greedy randomized construction. (a) Problem information; (b) giant tour; (c) auxiliary graph; (d) TTRP solution.

the route it represents $w_{i-1,j}=c(R_{ij})$, the consumption of trucks $\alpha_{i-1,j}$, and the consumption of trailers $\beta_{i-1,j}$. The quantities $\alpha_{i-1,j}$ and $\beta_{i-1,j}$ depend on the type of route in the following way: $\alpha_{i-1,j}=1, \beta_{i-1,j}=0$ if R_{ij} is a pure truck route, while $\alpha_{i-1,j}=\beta_{i-1,j}=1$ if R_{ij} is a pure vehicle route or a vehicle route with subtours.

In general, shortest path problems with resource constraints can be solved using a generalization of Bellman's algorithm with several labels per node [10]. In our case, let, $\Lambda=(\delta,\tau,\rho,\eta,\lambda)$ be a label associated with any given node $i \in X$ that represents a partial shortest path ending at node i. The label has five attributes: cost δ, truck consumption τ, trailer consumption ρ, father node η, and father label λ. Let $\mathcal{L}_i$ be the set of labels of node i, and let $\Gamma(i)$ be the set of successors of node i, $\Gamma(i)=\{j \in X : (i,j) \in U\}$.

For two labels $\Lambda_1=(\delta_1,\tau_1,\rho_1,\eta_1,\lambda_1)$, $\Lambda_2=(\delta_2,\tau_2,\rho_2,\eta_2,\lambda_2)$, we say that Λ_1 dominates in the Pareto sense Λ_2 (denoted $\Lambda_1 \preceq \Lambda_2$) if and only if $\delta_1 \le \delta_2 \wedge \tau_1 \le \tau_2 \wedge \rho_1 \le \rho_2$, and at least one of the inequalities is strict [15]. That is, label Λ_2 is dominated by label Λ_1 because it is possible to reach node j with the same distance and less resource consumption, or with a shorter distance and the same resource consumption.

We use Algorithm 1 to solve the RCSPP. Since by construction H is acyclic, the outmost *for* loop takes the nodes in increasing order, and for a given node i, it scans the set of successors $\Gamma(i)$ (lines 3–23). In the inmost *forall* loop (lines 4–22) all the labels of node i are extended, lines 11–17 perform a domination test, and remove all dominated labels of $\mathcal{L}_j$ if any exists. Finally, line 19 adds non-dominated labels to $\mathcal{L}_j, (j \in \Gamma(i))$. The number of arcs in U is bounded by $O(n^2)$. If we assume that there are no two labels with the same distance, the maximum number of non-dominated labels for each node can be bounded by $O(m_t m_r)$ because m_t and m_r are integers and the consumption is done one unit at a time. Thus, the non-domination test of lines 11–17 is performed in the worst case $O(m_t^2 m_r^2)$ times for each arc. The previous arguments prove that Algorithm 1 runs in $O(n^2\ m_t^2\ m_r^2)$.

Algorithm 1. Labeling algorithm for the resource-constrained shortest path problem

Input: Auxiliary graph H
Output: Shortest path from node 0 to n
1: Create a label $\Lambda_0=(0,0,0,0,\emptyset)$; $\mathcal{L}_0 := \mathcal{L}_0 \cup \{\Lambda_0\}$
2: **for** i=0 to $n-1$ **do**
3: **for all** $j \in \Gamma(i)$ **do**
4: **for all** $\overline{\Lambda}=(\overline{\delta},\overline{\tau},\overline{\rho},\overline{\eta},\overline{\lambda}) \in \mathcal{L}_i$ **do**
5: $ld := \overline{\delta}+w_{ij}$
6: $lt := \overline{\tau}+\alpha_{ij}$
7: $lr := \overline{\rho}+\beta_{ij}$
8: **if** $lt \le m_t$ and $lr \le m_r$ **then**
9: Create a label $\hat{\Lambda} := (ld,lt,lr,i,\overline{\Lambda})$
10: *nondom* := *true*
11: **for all** $\Lambda \in \mathcal{L}_j$ **do**
12: **if** $\hat{\Lambda} \preceq \Lambda$ **then**
13: $\mathcal{L}_j := \mathcal{L}_j \backslash \{\Lambda\}$
14: **else if** $\Lambda \preceq \hat{\Lambda}$ **then**
15: *nondom* := *false*
16: **end if**
17: **end for**
18: **if** *nondom* **then**
19: $\mathcal{L}_j := \mathcal{L}_j \cup \{\hat{\Lambda}\}$
20: **end if**
21: **end if**
22: **end for**
23: **end for**
24: **end for**

After solving the RCSPP, it is possible to derive the minimum-cost TTRP solution by selecting the label $\Lambda^* = \text{argmin}_{\Lambda \in \mathcal{L}_n} \delta_\Lambda$, (i.e., the cheapest label of node n) and backtracking from it using the information stored in η_{Λ^*} and λ_{Λ^*}. However, the algorithm for the RCSPP may fail to find a feasible solution and in that case $\mathcal{L}_n=\emptyset$. This occurs when it is not possible to reach node n with at most m_t trucks and m_r trailers. In this case, we relax the fleet-size constraints and solve a classical shortest path problem to find an infeasible solution. The infeasibility of the resulting solution is treated later in the improvement phase and the path relinking procedure.

Fig. 2 illustrates the greedy randomized construction for the TTRP. For the sake of clarity we only include in the auxiliary graph the cost of the arcs in the shortest path. The length of each square side in the grid is equal to 1 and the distance between nodes is Euclidean. All customers have unitary demands, except customer 1 with q_1=2. Using the information of the problem (Fig. 2(a)) and the sequence of the giant tour (Fig. 2(b)), the route-first cluster-second procedure first builds the auxiliary graph (Fig. 2(c)). After solving the shortest path problem from node 0 to node 1 in this graph, the route-first, cluster-second procedure builds the TTRP solution of Fig. 2(d) using the information of the arcs in the shortest path. Table 1 gives the details of each arc in the auxiliary graph including its tail and head, and the information of the associated route.

3.2. Variable neighborhood search for the TTRP

The improvement phase of the hybrid metaheuristic is a VNS procedure [25]. Taking an initial solution S_0, VNS performs a classical *variable neighborhood descent* (hereafter VND) step, and then repeats *ni* main iterations of shaking and improvement alternating between solutions and giant tours. Within VNS we accept infeasible solutions, provided that its infeasibility $\Phi(S)$ does not exceed a given threshold μ. The infeasibility of a given TTRP solution S is calculated using the following expression:

$$\Phi(S)=\max\left\{0,\frac{ut(S)}{m_t}-1\right\}+\max\left\{0,\frac{ur(S)}{m_r}-1\right\}$$

where $ut(S)$ and $ur(S)$ represent the number of trucks and trailers required by S. At each call of VNS, the value of μ is initialized at μ_{max} and decreased after each iteration by $\mu=\mu-\mu_{max}/ni$.

Let $T(S)$ be the giant tour associated with a given TTRP solution S, $T(S)$ is obtained by concatenating all the routes of S in a single string. The shaking procedure works on $T(S)$ by randomly exchanging b pairs of customers with procedure *perturb*($T(S)$,b). Then, we derive a new TTRP solution from the perturbed giant tour by using the RFCS approach described above. The value of b is controlled dynamically between 1 and b_{max}, depending on the feasibility and the objective function of the current solution. If the current solution is feasible and updates the best solution visited so far, the value of b is reset to 1 to search in its neighborhood. Whereas, if the current solution is infeasible or the best solution is not improved, b is increased to search in regions far from it. With this mechanism VNS acts as a reparation operator for infeasible solutions and as an improvement procedure for feasible ones.

On the other hand, the procedure *VND*(S) explores sequentially the following five neighborhoods of a given TTRP solution S using a best-improvement strategy:

- *Modified Or-opt*. For a given chain of customers $(r_i,\ldots,r_{i+l-1})$ of length $l=1,2,3$, check all possible reinsertions of the chain and its reverse $(r_{i+l-1},\ldots,r_i)$ within the same route or subtour. The difference with classical Or-opt is the simultaneous evaluation of the reversal of the chain.
- *Node exchange* (in single routes/subtours and between pairs of routes/subtours). Given a pair of customers u and v served by routes (or subtours) R_u and R_v exchange their positions. If $R_u \neq R_v$, in addition to classical capacity constraints, it is

Table 1
Arc information of the auxiliary graph for the route-first cluster-second example of Fig. 2.

Arc		Route				
Tail	Head	Type	Capacity	Load	Cost	Structure
0	4	PTR	2	1	2.00	0–4–0
0	3	PTR	2	2	3.41	(*) 0–4–3–0
4	3	PTR	2	1	2.83	0–3–0
4	7	PTR	2	2	4.65	0–3–7–0
4	6	VRWS	3	3	7.48	0–3–7–0 (Main tour) 7–6–7 (Subtour)
3	7	PTR	2	1	2.00	0–7–0
3	6	PTR	2	2	4.65	0–7–6–0
3	5	VRWS	3	3	6.83	(*) 0–7–0 (Main tour) 7–6–5–7 (Subtour)
7	6	PTR	2	1	4.47	0–6–0
7	5	PTR	2	2	5.89	0–6–5–0
6	5	PTR	2	1	4.47	0–5–0
6	2	PTR	2	2	8.94	0–5–2–0
5	2	PTR	2	1	4.47	0–2–0
5	1	PVR	3	3	6.47	(*) 0–2–1–0
2	1	PTR	2	2	4.47	0–1–0

PTR: pure truck route, PVR: pure vehicle route, VRWS: vehicle route with subtours,
(*): routes of the optimal splitting.

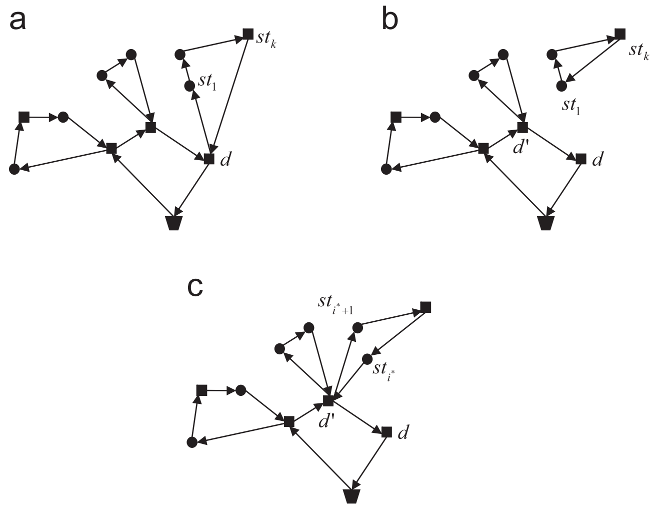

Fig. 3. Example of the root refining neighborhood [7]. (a) Initial Subtour. (b) Subtour as TSP. (c) Best insertion of the new root.

necessary to verify the accessibility constraints of u in R_v and v in R_u. Moreover when R_u or R_v is a subtour the capacity of the associated vehicle route is also checked.

- *2-opt* (in single routes/subtours, or pairs of routes (subtours) of the same type). Remove a pair of arcs (u,v) and (w,y) and add two other arcs. For single routes add arcs (u,w) and (v,y); if the arcs belong to different routes we also consider the addition of arcs (u,y) and (w,v) and select the best of the two options. Moreover, if the arcs belong to a pair of subtours we allocate the resulting subtours to the best root among those of the original subtours, provided the capacities of the associated vehicle routes are not exceeded.
- *Node relocation* (in single routes/subtours and between pairs of routes/subtours). Given a customer u served in route/subtour R_u and two consecutive nodes v,w in a route/subtour R', insert u between v and w. If $R_u \neq R'$ we check the conditions for a valid insertion of u in R'.
- *Root refining*. For each subtour we apply the root refining procedure described by Chao [7], where we try to change the root of each subtour and simultaneously modify its routing. Formally, for a subtour $ST=(d,st_1,\ldots,st_k,d)$ the operator removes arcs (d,st_1) and (st_k,d) and adds arc (st_k, st_1) to create a TSP tour. Then the position i^* of the new root d' is found using best insertion and the new subtour becomes $ST' = (d',st_{i^*+1},\ldots,st_1,st_k,\ldots,st_{i^*},d')$. To be feasible the new root d' must be served in pure vehicle routes or main tours of vehicle routes with subtours having enough residual capacity to insert the total demand of the subtour. Fig. 3 illustrates this neighborhood.

Using the elements described above, Algorithm 2 outlines the VNS component of the proposed metaheuristic.

Algorithm 2. VNS for the TTRP

Input: Initial solution S_0, **parameters:** μ_{max}, b_{max}, ni
Output: Improved solution S^*

$S_0 := VND(S_0)$
$b := 1$
$\mu := \mu_{max}$
$S^* := \emptyset$
if $\Phi(S_0) = 0$ **then**
  $S^* := S_0$
end if
$S := S_0$
for $i=1$ to ni **do**
  $T' := perturb(T(S), b)$
  $S' := RFCS(T')$
  $S' := VND(S')$
  if $\Phi(S') \le \mu$ and $f(S') < f(S)$ **then**
    $S := S'$
  end if
  if $\Phi(S') = 0$ and $f(S') < f^*$ **then**
    $S^* := S'$
    $b := 1$
  else
    $b := \min\{b+1, b_{max}\}$
  end if
  $\mu := \mu - \frac{\mu_{max}}{ni}$
end for
return S^*

3.3. Path relinking

Path relinking (PR) was introduced in the context of tabu search (TS) as a mechanism to combine intensification and diversification

Forward path

	Giant tour	Distance to T(S_f)
T(S_0)	0 7 6 5 3 4 1 2 0	6
T_1	0 1 2 7 6 5 3 4 0	5
T_2	0 1 2 3 4 7 6 5 0	4
T_3	0 1 2 3 4 5 7 6 0	3
T_4	0 1 2 3 4 5 6 7 0	0
T(S_f)	0 1 2 3 4 5 6 7 0	

Backward path

	Giant tour	Distance to T(S_f)
T(S_f)	0 1 2 3 4 5 6 7 0	6
T_1	0 7 1 2 3 4 5 6 0	5
T_2	0 7 6 1 2 3 4 5 0	4
T_3	0 7 6 5 1 2 3 4 0	3
T_4	0 7 6 5 3 4 1 2 0	0
T(S_0)	0 7 6 5 3 4 1 2 0	

Shifting blocks

Fig. 4. Example of the path relinking operator.

[23]. PR generates new solutions by exploring trajectories connecting elite solutions previously produced during the search. Hybridizing PR with GRASP improves the performance of the latter by tackling the memory-less criticism faced by the basic GRASP scheme [42].

Even though the use of PR in metaheuristics for vehicle routing is rather scarce, it has been applied with relative success. Hybrid metaheuristics combining PR and other methods have been used to solve the classical VRP [27,51], the multi-objective dial-a-ride problem [34], the multi-compartment VRP [16], and the VRP with time windows [26], among others. Particularly, GRASP/PR hybrids have been used to solve different routing problems such as the capacitated location-routing problem [38], the team orienteering problem [52], a combined production-distribution problem [3], and the capacitated arc-routing problem with time windows [39], among others.

Resende and Ribeiro [41] give an overview of several ways on how to hybridize PR with GRASP. However, the distance measure, the management of the set of elite solutions, and the design of the PR operator are independent from the hybridization mechanism. A brief description of these components for the TTRP follows.

3.3.1. *Distance measure and pool management*

GRASP with PR maintains a pool of elite solutions $\boldsymbol{ES}$. For inclusion in $\boldsymbol{ES}$ a solution S must be better than the worst solution of the pool, but to preserve the diversity of $\boldsymbol{ES}$, the distance $d(\boldsymbol{ES},S)$ between S and the pool must be greater than a given threshold Δ, where $d(\boldsymbol{ES},S)=\min_{S' \in \boldsymbol{ES}}\{d(S,S')\}$. However, the latter condition is overridden when the best solution of $\boldsymbol{ES}$ is updated. With the same diversity objective, S replaces $S_w = min_{S' \in \boldsymbol{ES}: f'(S') > f'(S)} d(S',S)$; i.e., the closer solution in $\boldsymbol{ES}$ that is worst than S. Note that since the pool may contain infeasible solutions, to guide the search toward feasible solutions we use a modified objective function $f' = M_1 \cdot \max\{0, ut(S)-m_t\} + M_2 \cdot \max\{0, ur(S)-m_r\} + f(S)$, where M_1 and M_2 are real numbers such that $M_1 \gg M_2 \gg 0$, and $f(S)$ is the total distance of the routes of S.

Initially $\boldsymbol{ES}$ is filled with $|\boldsymbol{ES}|$ solutions generated with GRASP/VND and $\boldsymbol{ES}$ is always kept ordered according to $f'(S)$. Note that, values for M_1 and M_2 are not explicitly needed because the pool is lexicographically sorted using an order consistent with the one imposed by $f'(S)$. This lexicographic order gives priority to feasible solutions, among feasible solutions to those with smaller distances, and among infeasible solutions to those with smaller infeasibility with respect to the use of trucks, and then to those with smaller infeasibility with respect to the use of trailers.

Different metrics can be used to define distances between solutions of vehicle routing problems. For instance, Ho and Gendreau [27] count the number of differing edges; Sörensen and Sevaux [50] measure the similarity between trips using the edit distance and then solve a linear assignment problem to match the trips of the solutions. Then the cost of the assignment becomes the distance between solutions. Finally, in route-first cluster-second based metaheuristics it is possible to measure the distance between solutions using their corresponding giant tours [37]. In that case, different metrics for distance on permutations could be used [46,49]. Among them, we decided to use the distance for R-permutations [4], also known as the adjacency or broken-pairs distance.

Given two solutions S and S' and their corresponding giant tours $T(S)$ and $T(S')$, the broken-pairs distance counts the number of consecutive pairs that differ from one giant tour to the other, that is $d(T(S),T(S'))$ is defined as the number of times t_{i+1} does not immediately follow t_i in $T(S')$, for $i=0,\ldots,|N|$. For example, if we have $T(S) = (0,1,2,3,4,5,0)$ and $T(S') = (0,5,3,4,1,2,0)$, the distance $d(T(S),T(S'))$ is 4, because pairs (0,1), (2,3), (4,5) and (5,0), of $T(S)$ are not in $T(S')$.

3.3.2. *Path relinking operator*

To transform the initial solution S_0 into the guiding solution S_f, the PR operator repairs from left to right the broken pairs of $T(S_0)$, creating a path of giant tours with non-increasing distance to $T(S_f)$. A broken pair is repaired by shifting to the left (in $T(S_0)$) blocks of consecutive customers in such a way that at least one broken pair is repaired without creating new broken pairs. Fig. 4 illustrates the PR operator. To increase the chance of finding high quality solutions, the PR operator uses the back-and-forward strategy [41], exploring the forward path from S_0 to S_f, and also the backward path from S_f to S_0. All the giant tours in both paths are split using the route-first cluster-second approach described above to generate a set $\boldsymbol{P}$ of TTRP solutions.

3.3.3. *Path relinking strategies and overview of the method*

Originally, Laguna and Martí [29] proposed PR as intensification mechanism after each GRASP iteration. Our first GRASP/VNS with PR described in Algorithm 3 follows this approach. In this hybrid method the PR operator explores the paths between a local optimum obtained by GRASP/VNS and a solution randomly chosen from $\boldsymbol{ES}$. The difference with the classical approach is that we apply VND to all feasible solutions produced by the PR operator and test them for insertion in $\boldsymbol{ES}$.

Algorithm 3. GRASP/VNS with PR as intensification mechanism

Input: TTRP, **parameters:** κ, ns, μ_{max},b_{max},ni, $|\boldsymbol{ES}|$, Δ
Output: TTRP solution S^*
for $i=1$ to $|\boldsymbol{ES}|$ **do**
  $T := RandomizedNearestNeighbor(N,\kappa)$
  $S := RFCS(T)$
  $S := VND(S)$
  Insert S in $\boldsymbol{ES}$
end for
for $i=1$ to ns **do**
  $T := RandomizedNearestNeighbor(N,\kappa)$

$S := RFCS(T)$
$S := VNS(S,\mu_{max},b_{max},ni)$
Select at random $S' \in \boldsymbol{ES}$
$\boldsymbol{P} := PathRelinkingOperator(S,S')$
for all $\bar{S} \in \boldsymbol{P} : \Phi(\bar{S}) = 0$ **do**
$\bar{S} := VND(\bar{S})$
if $d(\boldsymbol{ES},\bar{S}) \geq \Delta$ **then**
Try to insert $\bar{S}$ in $\boldsymbol{ES}$
end if
end for
end for
$S^* := \text{argmin}_{S \in \boldsymbol{ES}} f(S)$
return S^*

Another possibility is to use the PR operator as post-optimization procedure, after the *ns* iterations of GRASP/VNS. In this case, each local optimum produced by GRASP/VNS is just checked for inclusion in ***ES***. After the main GRASP/VNS loop, the procedure *PathRelinking*(***ES***), applies PR to all pairs of elite solutions in ***ES*** not yet relinked. The resulting feasible solutions (***RS***) are further improved with VND and ***ES*** updated. The post-optimization procedure is iterated as long as there are new solutions in ***ES***. Algorithm 4 summarizes this variant of GRASP/VNS with PR.

Algorithm 4. GRASP/VNS with PR as post-optimization mechanism

Input: TTRP, **parameters:** κ, ns, μ_{max},b_{max},ni, $|\boldsymbol{ES}|$, Δ
Output: TTRP solution S^*
for $i=1$ to ns **do**
$T := RandomizedNearestNeighbor(N,\kappa)$
$S := RFCS(T)$
$S := VNS(S,\mu_{max},b_{max},ni)$
if $d(\boldsymbol{ES},S) \geq \Delta$ **then**
Try to insert S in $\boldsymbol{ES}$
end if
end for
$new := true$
repeat
$\mathbf{ES}_0 := \boldsymbol{ES}$
$\boldsymbol{RS} := PathRelinking(\boldsymbol{ES})$
for all $\bar{S} \in \boldsymbol{RS}$ **do**
if $\Phi(\bar{S}) = 0$ **then**
$\bar{S} := VND(\bar{S})$
else
$\boldsymbol{RS} := \boldsymbol{RS} \setminus \{\bar{S}\}$
end if
end for
$\mathbf{ES} := Update(\mathbf{ES},\mathbf{RS})$
if $\mathbf{ES} \setminus \mathbf{ES}_0 = \emptyset$ **then**
$new := false$
end if
until $new = false$
$S^* := \text{argmin}_{S \in \boldsymbol{ES}} f(S)$
return S^*

More recently, Resende and Werneck [44] and Resende et al. [40] introduced evolutionary path relinking (EvPR), a variant in which the *PathRelinking*(***ES***), procedure is used periodically during the search. Consequently, our GRASP/VNS with EvPR for the TTRP (outlined in Algorithm 5) keeps the intensification mechanism, and evolves the elite set every γ iterations.

Algorithm 5. GRASP/VNS with evolutionary path relinking

Input: TTRP, **parameters:** κ, ns, μ_{max},b_{max},ni, $|\boldsymbol{ES}|$, Δ
Output: TTRP solution S^*
for $i=1$ to $|\boldsymbol{ES}|$ **do**
$T := RandomizedNearestNeighbor(N,\kappa)$
$S := RFCS(T)$
$S := VND(S)$
Insert S in $\boldsymbol{ES}$
end for
for $i=1$ to ns **do**
$T := RandomizedNearestNeighbor(N,\kappa)$
$S := RFCS(T)$
$S := VNS(S,\mu_{max},b_{max},ni)$
Select at random $S' \in \boldsymbol{ES}$
$\boldsymbol{P} := PathRelinkingOperator(S,S')$
for all $\bar{S} \in \boldsymbol{P} : \Phi(\bar{S}) = 0$ **do**
$\bar{S} := VND(\bar{S})$
if $d(\boldsymbol{ES},\bar{S}) \geq \Delta$ **then**
Try to insert $\bar{S}$ in $\boldsymbol{ES}$
end if
end for
if $i \bmod \gamma = 0$
$new := true$
repeat
$\mathbf{ES}_0 := \boldsymbol{ES}$
$\boldsymbol{RS} := PathRelinking(\boldsymbol{ES})$
for all $\bar{S} \in \boldsymbol{RS}$ **do**
if $\Phi(\bar{S}) = 0$ **then**
$\bar{S} := VND(\bar{S})$
else
$\boldsymbol{RS} := \boldsymbol{RS} \setminus \{\bar{S}\}$
end if
end for
$\mathbf{ES} := Update(\mathbf{ES},\mathbf{RS})$ **then**
if $\mathbf{ES} \setminus \mathbf{ES}_0 = \emptyset$ **then**
$new := false$
end if
until $new = false$
end if
end for
$S^* := \text{argmin}_{S \in \boldsymbol{ES}} f(S)$
return S^*

4. Computational experiments

We implemented the three variants of the proposed metaheuristic (GRASP/VNS with PR as post-optimization, GRASP/VNS with PR as intensification and GRASP/VNS with EvPR) using Java and compiled them using Eclipse JDT 3.5.1. We ran the experiments of this section on a computer with an Intel Xeon running at 2.67 GHz under Windows 7 Enterprise Edition (64 bits) with 4 GB of RAM. Table 2 summarizes the characteristics of each problem in the 21-instance test bed described by Chao [7], where the size of the problems range from $n = 50$ to 199 and for each problem size there are three values for the fraction of truck customers (25%, 50% and 75%).

All the variants of GRASP/VNS with PR share the size of the RCL (κ) and the number of iterations (ns) of GRASP; the maximum infeasibility threshold (μ_{max}), the maximum number of pairs (b_{max}) and the number of iterations (ni) of VNS; and the distance threshold (Δ) and size of the elite set ($|\mathbf{ES}|$) of PR. Additionally, EvPR is applied

Table 2
Test problems for the TTRP.

Problem number	Customers			Trucks		Trailers		Demand–capacity ratio
	n	$\|N_v\|$	$\|N_t\|$	m_t	Q_t	m_r	Q_r	
1	50	38	12					
2	50	25	25	5	100	3	100	0.971
3	50	13	37					
4	75	57	18					
5	75	38	37	9	100	5	100	0.974
6	75	19	56					
7	100	75	25					
8	100	50	50	8	150	4	100	0.911
9	100	25	75					
10	150	113	37					
11	150	75	75	12	150	6	100	0.931
12	150	38	112					
13	199	150	49					
14	199	100	99	17	150	9	100	0.923
15	199	50	149					
16	120	90	30					
17	120	60	60	7	150	4	100	0.948
18	120	30	90					
19	100	75	25					
20	100	50	50	10	150	5	100	0.903
21	100	25	75					

Table 3
Parameters of the proposed GRASP/VNS with PR variants.

Method	GRASP		VNS			Path relinking		
	κ	ns	μ_{max}	b_{max}	ni	$\|\mathbf{ES}\|$	Δ	γ
GRASP/VNS	2	60	0.25	6	200	–	–	–
GRASP/VNS with PR (intensification)	2	60	0.25	6	200	5	$\max\{10, m_t+m_r\}$	–
GRASP/VNS with PR (post-optimization)	2	60	0.25	6	200	5	$\max\{10, m_t+m_r\}$	–
GRASP/VNS with EvPR	2	60	0.25	6	200	5	$\max\{10, m_t+m_r\}$	20

Table 4
Results of the proposed metaheuristics in the test problems of [7].

Problem			GRASP/VNS			GRASP/VNS +PR (intensification)			GRASP/VNS +PR (post-optimization)			GRASP/VNS + EvPR		
Number	n	BKS	Best	Avg.	Time	Best	Avg.	Time	Best	Avg.	Time	Best	Avg.	Time
1	50	564.68	**564.68**	568.31	0.91	**564.68**	566.22	1.07	**564.68**	566.38	0.98	**564.68**	565.99	1.17
2	50	611.53	614.27	617.53	1.00	**611.53**	614.46	1.18	**611.53**	614.81	1.08	**611.53**	614.23	1.29
3	50	618.04	**618.04**	619.07	0.86	**618.04**	618.04	0.98	**618.04**	618.24	0.91	**618.04**	618.04	1.05
4	75	798.53	802.41	815.16	1.86	802.41	805.72	2.37	799.34	805.62	2.08	**798.53**	803.51	2.69
5	75	839.62	841.81	857.79	1.99	**839.62**	844.99	2.59	840.74	848.18	2.19	**839.62**	841.63	2.82
6	75	930.64	989.71	1040.19	1.95	952.43	967.77	2.57	946.66	970.18	2.36	940.59	961.47	2.89
7	100	830.48	830.62	832.27	4.11	**830.48**	830.55	5.55	**830.48**	830.71	4.66	**830.48**	830.48	6.05
8	100	872.56	881.53	885.01	4.35	874.95	878.48	5.99	874.73	879.02	5.12	**872.56**	876.21	6.96
9	100	912.02	916.63	930.55	5.67	915.29	920.22	7.66	915.46	921.39	6.25	914.23	918.45	8.38
10	150	1039.07	1050.76	1062.03	9.95	1047.25	1051.40	16.01	1047.59	1054.40	13.10	1046.71	1050.11	18.84
11	150	1093.37	1114.64	1122.43	11.02	1095.94	1108.45	18.30	1097.75	1105.37	14.63	**1093.37**	1100.95	21.20
12	150	1152.32	1159.88	1174.89	14.00	1155.09	1163.67	23.21	1153.04	1159.11	18.44	**1152.32**	1158.88	25.78
13	199	1287.18	1319.38	1332.55	18.71	1304.77	1314.52	36.33	1301.22	1310.78	26.15	1298.89	1305.83	43.94
14	199	1339.36	1380.86	1395.50	20.07	1357.05	1367.50	39.96	1351.23	1362.02	30.02	**1339.36**	1354.04	45.57
15	199	1420.72	1454.10	1462.23	25.14	1430.38	1443.45	52.53	**1420.72**	1436.29	37.12	1423.91	1437.52	59.83
16	120	1002.49	1003.99	1005.88	8.13	**1002.49**	1003.51	12.12	**1002.49**	1003.82	9.99	**1002.49**	1003.07	14.73
17	120	1026.20	1045.08	1050.86	8.09	1042.53	1042.99	11.86	1042.53	1044.76	9.41	1042.46	1042.61	13.17
18	120	1098.15	1121.07	1128.51	7.73	1114.33	1121.00	11.16	1113.18	1120.02	9.01	1113.07	1118.63	12.69
19	100	813.30	817.11	820.94	3.82	814.73	820.45	4.74	813.72	820.35	4.18	813.50	819.81	5.21
20	100	848.93	860.12	861.34	4.21	860.12	860.12	5.28	860.12	860.12	4.49	860.12	860.12	5.62
21	100	909.06	912.35	913.62	4.59	**909.06**	909.60	5.83	**909.06**	910.33	5.06	**909.06**	909.06	6.31
Avg. gap above BKS			1.29%	2.26%		0.60%	1.11%		0.48%	1.09%		0.36%	0.84%	
NBKS			2			7			7			12		
Avg. time (min)					7.53			12.73			9.87			14.58

Table 5
Comparison of GRASP/VNS with EvPR against other approaches from the literature.

Problem				Chao [7] (tabu search)		Scheuerer [45] (tabu search)				Lin et al. [32] (simulated annealing)				Caramia and Guerriero [5] (math. prog. heuristic)		GRASP/VNS + EvPR			
Number	*n*	BKS	Reference	Avg. Cost	Gap	Best cost	Gap	Avg. Cost	Gap	Best cost	Gap	Avg. Cost	Gap	Cost	Gap	Best cost	Gap	Avg. Cost	Gap
1	50	564.68	[45]	565.02	0.06	566.80	0.38	567.98	0.59	566.82	0.38	568.86	0.74	566.80	0.38	**564.68**	0.00	565.99	0.23
2	50	611.53	[32]	662.84	8.39	615.66	0.68	619.35	1.28	612.75	0.20	617.48	0.97	620.15	1.41	**611.53**	0.00	614.23	0.44
3	50	618.04	[45]	664.73	7.56	620.78	0.44	629.59	1.87	**618.04**	0.00	620.50	0.40	632.48	2.34	**618.04**	0.00	618.04	0.00
4	75	798.53	[45]	857.84	7.43	801.60	0.38	809.13	1.33	808.84	1.29	817.71	2.40	803.32	0.60	**798.53**	0.00	803.51	0.62
5	75	839.62	[45]	949.98	13.14	**839.62**	0.00	858.98	2.31	**839.62**	0.00	858.95	2.30	842.50	0.34	**839.62**	0.00	841.63	0.24
6	75	930.64	[32]	1084.82	16.57	936.01	0.58	949.89	2.07	934.11	0.37	942.60	1.29	938.18	0.81	940.59	1.07	961.47	3.31
7	100	830.48	[45]	837.80	0.88	**830.48**	0.00	832.91	0.29	**830.48**	0.00	838.50	0.97	832.56	0.25	**830.48**	0.00	830.48	0.00
8	100	872.56	[32]	906.16	3.85	878.87	0.72	881.26	1.00	875.76	0.37	882.70	1.16	878.87	0.72	**872.56**	0.00	876.21	0.42
9	100	912.02	[32]	1000.27	9.68	942.31	3.32	955.95	4.82	912.64	0.07	921.97	1.09	980.42	7.50	914.23	0.24	918.45	0.71
10	150	1039.07	[45]	1076.88	3.64	1039.23	0.02	1052.65	1.31	1053.90	1.43	1074.38	3.40	1060.41	2.05	1046.71	0.74	1050.11	1.06
11	150	1093.37	(a)	1170.17	7.02	1098.84	0.50	1107.47	1.29	1093.57	0.02	1108.88	1.42	1170.70	7.07	**1093.37**	0.00	1100.95	0.69
12	150	1152.32	(a)	1217.01	5.61	1175.23	1.99	1184.58	2.80	1155.44	0.27	1166.59	1.24	1178.34	2.26	**1152.32**	0.00	1158.88	0.57
13	199	1287.18	[45]	1364.50	6.01	1288.46	0.10	1296.33	0.71	1320.21	2.57	1340.98	4.18	1288.46	0.10	1298.89	0.91	1305.83	1.45
14	199	1339.36	(a)	1464.20	9.32	1371.42	2.39	1384.13	3.34	1351.54	0.91	1367.91	2.13	1372.52	2.48	**1339.36**	0.00	1354.04	1.10
15	199	1420.72	(b)	1544.21	8.69	1459.55	2.73	1488.71	4.79	1436.78	1.13	1454.91	2.41	1470.21	3.48	1423.91	0.22	1437.52	1.18
16	120	1002.49	[45]	1064.89	6.22	**1002.49**	0.00	1003.00	0.05	1004.47	0.20	1007.26	0.48	1004.69	0.22	**1002.49**	0.00	1003.07	0.06
17	120	1026.20	[32]	1104.67	7.65	1042.35	1.57	1042.79	1.62	1026.88	0.07	1035.23	0.88	1042.35	1.57	1042.46	1.58	1042.61	1.60
18	120	1098.15	[32]	1202.00	9.46	1129.16	2.82	1141.94	3.99	1099.09	0.09	1110.13	1.09	1129.16	2.82	1113.07	1.36	1118.63	1.86
19	100	813.30	[32]	887.22	9.09	813.50	0.02	813.98	0.08	814.07	0.09	823.01	1.19	813.50	0.02	813.50	0.02	819.81	0.80
20	100	848.93	[45]	963.06	13.44	**848.93**	0.00	852.89	0.47	855.14	0.73	859.06	1.19	**848.93**	0.00	860.12	1.32	860.12	1.32
21	100	909.06	[45]	952.29	4.76	**909.06**	0.00	914.04	0.55	**909.06**	0.00	915.38	0.70	**909.06**	0.00	**909.06**	0.00	909.06	0.00
Average cost				1025.74				970.84				968.24		970.65				961.46	
Avg. gap above BKS					7.55%		0.89%		1.74%		0.48%		1.51%		1.74%		0.36%		0.84%
NBKS						5				4				2		12			

(a) GRASP/VNS + EvPR, (b) GRASP/VNS with PR (post-optimization).

Table 6
Data for the Friedman test on the ranking of the average performance of each metaheuristic.

Problem number	Chao [7]		Scheuerer [45]		Lin et al. [32]		GRASP/VNS + EvPR	
	Avg. Cost	Rank	Avg. cost	Rank	Avg. cost	Rank	Avg. cost	Rank
1	565.02	1	567.98	3	568.86	4	565.99	2
2	662.84	4	619.35	3	617.48	2	614.23	1
3	664.73	4	629.59	3	620.50	2	618.04	1
4	857.84	4	809.13	2	817.71	3	803.51	1
5	949.98	4	858.98	3	858.95	2	841.63	1
6	1084.82	4	949.89	2	942.60	1	961.47	3
7	837.80	3	832.91	2	838.50	4	830.48	1
8	906.16	4	881.26	2	882.70	3	876.21	1
9	1000.27	4	955.95	3	921.97	2	918.45	1
10	1076.88	4	1052.65	2	1074.38	3	1050.11	1
11	1170.17	4	1107.47	2	1108.88	3	1100.95	1
12	1217.01	4	1184.58	3	1166.59	2	1158.88	1
13	1364.50	4	1296.33	1	1340.98	3	1305.83	2
14	1464.20	4	1384.13	3	1367.91	2	1354.04	1
15	1544.21	4	1488.71	3	1454.91	2	1437.52	1
16	1064.89	4	1003.00	1	1007.26	3	1003.07	2
17	1104.67	4	1042.79	3	1035.23	1	1042.61	2
18	1202.00	4	1141.94	3	1110.13	1	1118.63	2
19	887.22	4	813.98	1	823.01	3	819.81	2
20	963.06	4	852.89	1	859.06	2	860.12	3
21	952.29	4	914.04	2	915.38	3	909.06	1
Average Rank		3.81		2.29		2.43		1.48
Sum of Ranks		80		48		51		31
Squared Sum of Ranks		6400		2304		2601		961

every γ iterations. We also included in the computational experiment a GRASP/VNS (without PR) as a base case (benchmark) to analyze the contributions of PR. After some preliminary experimentation we set the parameters of the different variants of the hybrid metaheuristic to the values summarized in Table 3.

Table 4 presents the best and average results over 10 runs of the GRASP/VNS with PR variants and the GRASP/VNS benchmark. The column labeled *Time* reports the average running time in minutes for each method. We also include the best-known solutions (*BKS*) for each instance, taken from Lin et al. [32] and Scheuerer [45] and updated with some new best-known solutions found by the proposed GRASP/VNS with PR. The last rows of the table summarize the average gap above best-known solutions, the number of times each method found the best-known solution (*NBKS*), and the average running time. Values in bold in the table indicate that the BKS was found by a given method.

All the variants of GRASP/VNS with PR largely outperform the base case GRASP/VNS (without PR), highlighting the contribution of PR to the quality of solutions. Remarkably, the use of PR as a post-optimization mechanism offers a good trade-off between running time and solution quality. The post-optimization with PR approximately halved the average gap to BKS of GRASP/VNS with an increase of only 30% in the running time. Moreover, this variant was able to improve the BKS of problem 15.[1] Above all, GRASP/VNS with EvPR stands as the best performing method, having an average gap to BKS as small as 0.84% obtaining 12 out of 21 BKS, and improving the BKS of problems 11, 12 and 14.[1] However, this outstanding performance is achieved at the price of almost doubling the running time of the benchmark GRASP/VNS.

Table 5 presents the comparison of the proposed hybrid metaheuristic against other methods from the literature. In this table we only compare against the best variant, namely GRASP/VNS with EvPR. Table 4 includes the results of the tabu search of Chao [7] and Scheuerer [45], the simulated annealing of Lin et al. [32], and the mathematical-programming-based heuristic of Caramia and Guerriero [5]. Depending of the availability of results we report the best and average results over 10 runs of each metaheuristic. For the heuristic of Caramia and Guerriero, we only report the results of a single run of their deterministic method. Column *BKS* presents the best known solution for each problem reported for the first time in the paper cited in column *Ref*, column *Gap* reports the gap with respect to BKS (in %) for each instance and each method. The last rows of Table 5 summarize the average cost over the 21 test problems, the average gap above best known solutions (*BKS*), and the number of times each method found the best-known solution (*NBKS*).

As can be seen in Table 5, GRASP/VNS with EvPR outperforms all the methods from the literature, achieving a small average gap to BKS of 0.84% and obtaining 12 out of 21 BKS with a single set of parameters. Our method improved the BKS for four large problems. GRASP/VNS with EvPR almost halved the average gap to BKS of the simulated annealing heuristic of Lin et al. [32], the previous best method with an average gap to BKS of 1.51%, and the second-best methods by Scheuerer [45] and Caramia and Guerriero [5], which obtained the same average gap to BKS of 1.74%. Finally, the tabu search of Chao [7] with an average gap to BKS of 7.55% is clearly outperformed by GRASP/VNS with EvPR.

It is important to note that the worst performance of EvPR (as for the other variants of the hybrid metaheuristic) is obtained on problem 6. By analyzing some statistics during the search, we observed that due to the very tight demand to capacity ratio (0.974) it is very difficult to find feasible solutions with the proposed RFCS approach. In contrast, the method by Caramia and Guerriero [5] is better adapted to solve this problem since it has a packing step that produces a feasible solution if any exists.

Some authors used the average cost over the 21 instances as a measure to compare the performance of different metaheuristics for the TTRP [5,32]. Nonetheless, this is not a good measure because it favors methods with good results in the larger test problems with overall large cost due solely to their size. For instance, the cost of the BKS of problem 15 is 2.5 times the cost of the BKS of problem 1. Then an improvement of 1% of the BKS of problem 15 will have 2.5 times more impact in this measure than the same relative

[1] Detailed solutions available at http://hdl.handle.net/1992/1127

improvement on problem 1. Hence, to have a better comparison of the algorithms, we followed Garcia et al. [19] and used the Friedman test to analyze the average results of the randomized metaheuristics.

The null hypothesis of the Friedman test is that each ranking of the algorithms within each problem is equally likely, so there is no difference between them. As can be seen in Table 6, GRASP/VNS with EvPR consistently ranks in the first two positions. The Friedman test was performed according to the procedure described by Conover [8] and the analysis led to the rejection of the null hypothesis with a level of significance $\alpha = 1\%$. Moreover, the paired comparisons unveiled that GRASP/VNS with EvPR is better than each one of the other methods with the same level of significance.

Since the method of Caramia and Guerriero [5] is deterministic, only one run is enough to characterize its performance. To have a fair comparison against it, in Table 7 we compared their results against those of the best and worst runs (i.e., the seeds that produce the smallest and biggest average deviations above BKS, respectively). In the

Table 7
Comparison of a single run of GRASP/VNS with EvPR and Caramia and Guerriero's heuristic.

Problem				Caramia and Guerrriero		Best seed		Worst seed	
Number	n	BKS	Reference	Cost	Gap	Cost	Gap	Cost	Gap
1	50	564.68	[45]	566.80	0.38	566.09	0.25	566.82	0.38
2	50	611.53	[32]	620.15	1.41	613.61	0.34	615.18	0.60
3	50	618.04	[45]	632.48	2.34	618.04	0.00	618.04	0.00
4	75	798.53	[45]	803.32	0.60	799.34	0.10	798.87	0.04
5	75	839.62	[45]	842.50	0.34	843.05	0.41	842.47	0.34
6	75	930.64	[32]	938.18	0.81	948.42	1.91	980.34	5.34
7	100	830.48	[45]	832.56	0.25	830.48	0.00	830.48	0.00
8	100	872.56	[32]	878.87	0.72	879.51	0.80	881.17	0.99
9	100	912.02	[32]	980.42	7.50	914.23	0.24	920.07	0.88
10	150	1039.07	[45]	1060.41	2.05	1051.70	1.22	1053.22	1.36
11	150	1093.37	(a)	1170.70	7.07	1106.41	1.19	1099.05	0.52
12	150	1152.32	(a)	1178.34	2.26	1152.32	0.00	1162.16	0.85
13	199	1287.18	[45]	1288.46	0.10	1309.69	1.75	1307.35	1.57
14	199	1339.36	(a)	1372.52	2.48	1347.61	0.62	1361.63	1.66
15	199	1420.72	(b)	1470.21	3.48	1424.31	0.25	1438.61	1.26
16	120	1002.49	[45]	1004.69	0.22	1002.82	0.03	1002.49	0.00
17	120	1026.20	[32]	1042.35	1.57	1042.53	1.59	1042.63	1.60
18	120	1098.15	[32]	1129.16	2.82	1113.07	1.36	1122.88	2.25
19	100	813.30	[32]	813.50	0.02	821.85	1.05	821.85	1.05
20	100	848.93	[45]	848.93	0.00	860.12	1.32	860.12	1.32
21	100	909.06	[45]	909.06	0.00	909.06	0.00	909.06	0.00
Avg. gap above BKS					1.74%		0.69%		1.05%
Number of times better than Caramia and Guerriero						13		13	

(a) GRASP/VNS + EvPR, (b) GRASP/VNS with PR (post-optimization).

Table 8
Comparison of the running time of GRASP/VNS with EvPR and other published methods.

Problem		GRASP/VNS with EvPR		Scheuerer [45]		Lin et al. [32]	
Number	n	Avg. time	Scaled time ($\times 2.15$)	Avg. time	Speed-up	Avg. time	Speed-up
1	50	1.17	2.51	9.51	3.79	6.80	2.71
2	50	1.29	2.77	9.60	3.47	6.67	2.41
3	50	1.05	2.27	11.24	4.95	5.59	2.46
4	75	2.69	5.79	18.49	3.20	16.32	2.82
5	75	2.82	6.07	15.16	2.50	14.42	2.37
6	75	2.89	6.22	18.62	2.99	13.65	2.19
7	100	6.05	13.02	33.60	2.58	24.96	1.92
8	100	6.96	14.97	25.66	1.71	24.03	1.61
9	100	8.38	18.03	30.47	1.69	21.75	1.21
10	150	18.84	40.54	60.94	1.50	63.61	1.57
11	150	21.20	45.62	56.17	1.23	60.33	1.32
12	150	25.78	55.49	63.71	1.15	51.70	0.93
13	199	43.94	94.56	165.41	1.75	119.56	1.26
14	199	45.57	98.08	132.06	1.35	113.75	1.16
15	199	59.83	128.76	154.10	1.20	93.87	0.73
16	120	14.73	31.69	43.14	1.36	41.46	1.31
17	120	13.17	28.35	33.73	1.19	38.81	1.37
18	120	12.69	27.31	31.78	1.16	31.34	1.15
19	100	5.21	11.21	28.84	2.57	29.58	2.64
20	100	5.62	12.09	24.57	2.03	28.47	2.36
21	100	6.31	13.58	26.84	1.98	24.03	1.77
Average		14.58	31.38	47.32		39.56	
Geometric mean					1.96		1.66

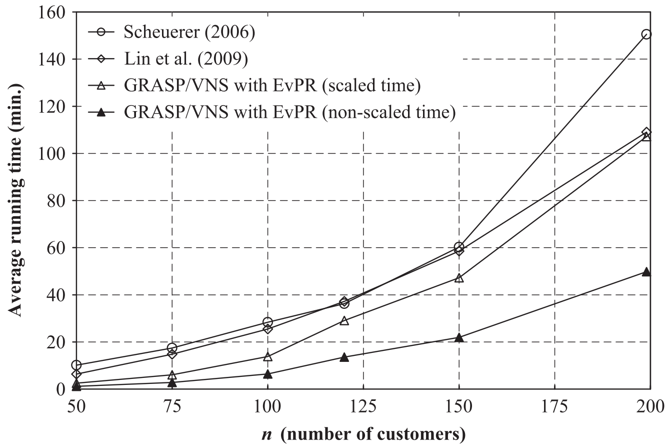

Fig. 5. Comparison of average running time of GRASP/VNS with EvPR and other published methods.

last row of Table 7 it is possible to see that GRASP/VNS with EvPR is consistently better than Caramia and Guerriero's method in 13 out of 21 problems, regardless of the seed. Even though, the best seed has a much smaller gap to BKS of 0.69% compared to 1.74% of their method, the worst seed is still better than their method with a gap to BKS of 1.05%. This experiment also highlights the robustness of GRASP/VNS with EvPR, that is, the performance of a single run is very close to the average performance.

To compare the running times of GRASP/VNS with EvPR against those reported in the literature we chose the tabu search of Scheuerer [45] and the simulated annealing of Lin et al. [32]. The tabu search of Chao [7] was discarded because it has a large gap to BKS and, unfortunately, Caramia and Guerriero [5] did not report running times.

To have a fair comparison of the running times we scaled the time spent by GRASP/VNS with EvPR to the reference computer used by Scheuerer [45] and Lin et al. [32]. Both of them used an Intel Pentium IV PC running at 1.5 GHz. Scheuerer [45] reported a speed factor of around 326 Mflop/s (millions of floating-point operations per second) for this computer using the Linpack benchmark [11]. Using the Java version of the Linpack benchmark [12], we obtained a speed factor of approximately 702 Mflop/s for our Intel Xeon running at 2.67 GHz. Using these values we derived a scaling factor of 2.15 for our running times.

Table 8 shows that GRASP/VNS with EvPR has in general shorter running times. The column labeled *Speed-up* reports the ratio of the running times of the published algorithms over the scaled time of GRASP/VNS with EvPR. Following the arguments outlined by Bixby [2], we report in the last row the geometric mean of these ratios as a conservative estimate of the speed-up factor achieved by GRASP/VNS with EvPR with respect to the tabu search of Scheuerer [45] and the simulated annealing of Lin et al. [32]. The comparison of Table 8 is further illustrated through Fig. 5. However, this running time comparison must be taken with care since the operating system, computer configuration, and programming language varies for each method.

5. Conclusions

In this paper we proposed a hybrid metaheuristic based on GRASP, VNS and path relinking to solve the truck and trailer routing problem. The constructive phase and the path relinking operator are based on a route-first, cluster-second approach. During the search the proposed metaheuristic explores infeasible solutions while the VNS component plays the role of repairing operator and improving mechanism. The computational experiments on a set of 21 standard test instances from the literature unveils the accuracy of the proposed GRASP/VNS with path relinking and the contribution of path relinking to the quality of solutions. Moreover, the proposed hybrid metaheuristic outperforms all previous published methods, and exhibits a very small variability when comparing the results of a single run against the average results over several runs.

After exploring different hybridization alternatives for the path relinking component, GRASP/VNS with evolutionary path relinking emerged as the overall winner, achieving a small average gap to best-known solutions of 0.84%, finding 12 out of 21 best known solutions, and improving 3 of them with a single set of parameters. The GRASP/VNS with path relinking as post-optimization mechanism variant is 32% faster than GRASP/VNS with evolutionary path relinking without a significant sacrifice on the quality of the results, achieving an average gap to best-known solution of 1.09%, and improving the best-known solution of one of the larger problems.

Further research directions include the development of lower bounds and exact methods to solve the TTRP, and the study of the multi-objective TTRP in which the fleet size (number of trucks and number of trailers), and the total distance are used as objective functions.

Acknowledgments

The authors would like to thank Maria Elsa Correal and Maria Isabel Restrepo at the Universidad de Los Andes for their statistical advice. This work was partly founded by the Champagne-Ardenne Regional Council (France) and by Colciencias, (Colombia). This support is gratefully acknowledged.

Appendix A. Notation

The symbols used throughout the paper are summarized in Tables A1–A4.

Appendix B. Solution of the restricted STTRPSD

The cost of vehicle routes with subtours is obtained by solving a restricted version of the STTRPSD (Villegas et al. [54]). Firstly, we recall the notation. The solution of the restricted STTRPSD is obtained with a dynamic programming method in which state $[l,m](i \le l \le j; l : t_l \in N_v; l \le m \le j)$ represents the use of vehicle customer t_l as the root of a subtour ending at customer t_m. F_{lm} denotes the cost of state $[l,m]$, and θ_{lkm} is the cost of a subtour $ST=(t_l,t_k,\ldots,t_m,t_l)$ rooted at vehicle customer t_l and visiting customer

Table A1
Notation for the definition of the TTRP.

Symbol	Description
m_t	Number of available trucks
m_r	Number of available trailers
Q_t	Truck capacity
Q_r	Trailer capacity
$N=\{1,\ldots,n\}$	Set of customers
N_v	Subset of vehicle customers
N_t	Subset of truck customers
q_i	Demand for customer i
c_{ij}	Distance between nodes i and j

Table A2
Notation for the route-first, cluster-second procedure.

Symbol	Description
$T=(0,t_1,\ldots,t_i,\ldots,t_n,0)$	Giant tour
$H=(X,U,W)$	Auxiliary graph for the tour splitting procedure
X	Set of nodes of the auxiliary graph
U	Set of arcs of the auxiliary graph
W	Weight of the arcs in the auxiliary graph
(i,j)	Arc of the auxiliary graph
w_{ij}	Cost of arc (i,j)
$R_{ij}=(0,t_i,\ldots,t_j,0)$	Route serving from customer t_i to customer t_j of giant tour T
$c(R_{ij})$	Cost (total distance) of route R_{ij}
Q_{ij}	Total demand of route R_{ij}
$[l,m]$	State of the dynamic programming method used for the cost of vehicle routes with subtours
F_{lm}	Cost of state $[l,m]$
θ_{lkm}	Cost of subtour $ST=(t_l,t_k,\ldots,t_m,t_l)$
α_{ij}	Truck consumption of arc $(i,j)\in U$
β_{ij}	Trailer consumption of arc $(i,j)\in U$
$\Lambda=(\delta,\tau,\rho,\nu,\lambda)$	Label for the solution of the resource-constrained shortest path problem
δ	Distance of label Λ
τ	Truck consumption of label Λ
ρ	Trailer consumption of label Λ
ν	Father node of label Λ
λ	Father label of label Λ
$\mathcal{L}_i$	Set of labels of node i in the auxiliary graph
$\Gamma(i)$	Set of successors of node i in the auxiliary graph

Table A3
Notation of VNS and path relinking.

Symbol	Description
S	TTRP solution
S_0	Initial solution
b	Number of pairs of customers exchanged in the perturbation procedure of VNS
$\Phi(S)$	Infeasibility of solution S
μ	Infeasibility threshold
$ut(S)$	Number of trucks used in solution S
$ur(S)$	Number of trailers used in solution S
$T(S)$	Giant tour of solution S
ES	Pool of elite solutions
$d(S,S')$	Distance between solutions S and S'
$d(\mathbf{ES},S)$	Distance between solution S and the pool **ES**
$f'(S)$	Modified objective function for the ordering of the pool
S_f	Final solution of the path relinking operator
P, **RS**	Sets of solutions produced by path relinking

t_k to t_m. It is possible to find the structure of the route and its cost using the following recurrence relation:

$$F_{lj}=\begin{cases} c_{0t_i} & \text{if } l=i \text{ and } m=i \\ \min\limits_{k<m:\sum_{u=k+1}^{m} q_{t_u}\le Q_t} \{F_{lk}+\theta_{i,k+1,m}\} & \text{if } l=i \text{ and } i<m\le j \\ \min\left\{ \min\limits_{k<m:\sum_{u=k+1}^{m} q_{t_u}\le Q_t} \{F_{lk}+\theta_{l,k+1,m}\},\ \min\limits_{\substack{h<l:t_h\in N_v,\\ k=l-1:\sum_{u=k+2}^{m} q_{t_u}\le Q_t}} \{F_{hk}+c_{t_h,t_l}+\theta_{l,k+2,m}\} \right\} & \text{if } i<l\le j: t_l\in N_v, \text{ and } l\le m\le j \end{cases}$$

Since states $[l,m]$, do not include the return from the last vehicle customer to the main depot, the cost of the route is finally calculated as $c(R_{ij})=\min_{i\le l\le j:t_l\in N_v}\{F_{lj}+c_{t_l,0}\}$.

Following the same approach as Villegas et al. [54], the dynamic programming method for the STTRPSD can be represented by an auxiliary graph $G=(V,A,Z)$. The node set V is composed of the nodes

Table A4
Parameters of the hybrid metaheuristic.

Symbol	Description
ns	Number of GRASP iterations
κ	Cardinality of the restricted candidate list of GRASP
ni	Number of VNS iterations
μ_{max}	Maximum infeasibility threshold of VNS
b_{max}	Maximum number of pairs for the shaking of VNS
Δ	Minimum distance threshold in path relinking
$\lvert\mathbf{ES}\rvert$	Cardinality of the pool of elite solutions
γ	Frequency of evolutionary path relinking

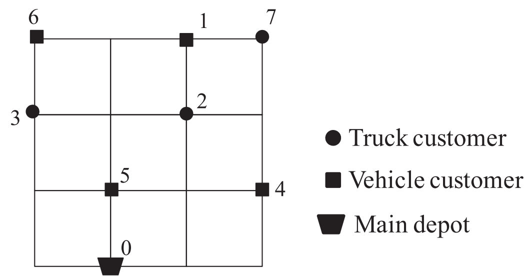

Fig. B1. Data for the solution of the restricted STTRPSD.

representing states $[l,m]$, and a dummy node ω representing the return to the main depot.

The arc set A contains three types of arcs. Arcs $([l,k],[l,m]), i\le l\le j, t_l\in N_v; k<m\le j$ represent a subtour that serves customers $(t_{k+1},\ldots,t_m)$ rooted at vehicle customer t_l without moving the trailer that is already parked at vehicle customer t_l; the cost of these arcs is given by $\theta_{l,k+1,m}$. Arcs of the form $([h,k],[l,m]), i\le h<l\le j, t_h,t_l\in N_v; k=l-1\le j:\sum_{u=k+2}^{m} q_{t_u}\le Q_t$ represent a subtour that serves customers $(t_{k+2},\ldots,t_m)$ rooted at vehicle customer t_l coming from vehicle customer t_h after performing the subtour that ends at customer t_k; the cost of these arcs is $c_{t_h,t_l}+\theta_{l,k+2,m}$. Finally, we have arcs of the form $([l,j],\omega), t_l\in N_v$ representing the return of the complete vehicle to the main depot after serving the last customer of the route in a subtour rooted at vehicle customer t_l. The cost of these arcs is $c_{t_l,0}$.

We obtain the structure of the route and its cost by finding in G the shortest path from state $[i,i]$ to node ω. As shown in Villegas et al. [54], this problem can be solved efficiently without generating explicitly the auxiliary graph G. A small example follows. The location of the customers and its type is given in Fig. B1, the length of each square side in the grid is equal to 1 and the distance between nodes is Euclidean.

Given a giant tour $T=(0,6,5,3,1,2,4,7,0)$ for a TTRP with $Q_r=2$, $Q_t=3$ and unitary demands. We are interested in the structure and cost of route $R_{2,6}=(0,5,3,1,2,4,0)$. The auxiliary graph G is given in Fig. B2(a), the arcs in bold correspond to the shortest path, and the associated solution is given in Fig. B2(b). Note that in the auxiliary graph we replaced state $[l,m]$ with $[t_l,t_m]$ to simplify the presentation.

Table B1 details the calculation of the cost of the arcs in the shortest path. Note that the cost of the route is 11.30, while the cost

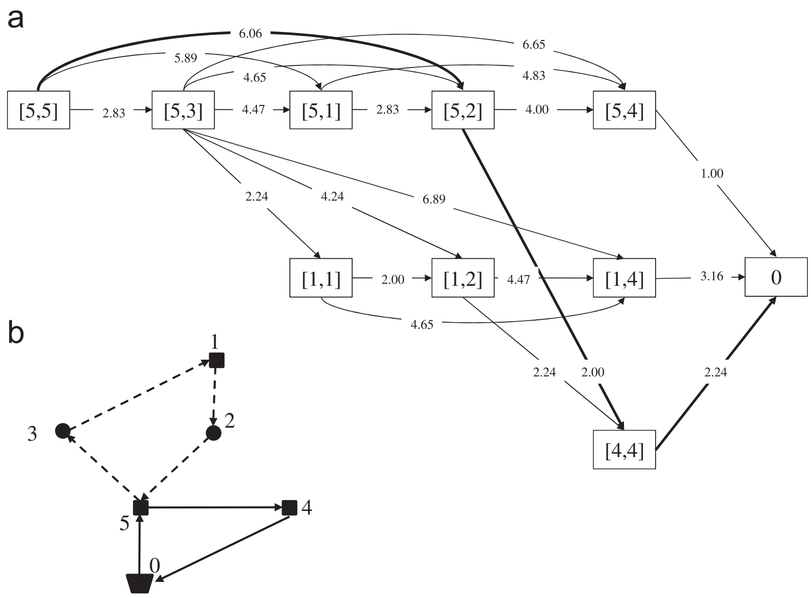

Fig. B2. Example of the restricted STTRPSD used to find the cost of vehicle routes with subtours: (a) Auxiliary graph. (b) Vehicle route with subtours.

Table B1
Example of the cost of the arcs of G for the STTRPSD.

Arc	Cost	Calculation	Value
([5,5],[5,2])	$\theta_{2,3,5}=c_{t_2,t_3}+c_{t_3,t_4}+c_{t_4,t_5}+c_{t_5,t_2}$	$c_{5,3}+c_{3,1}+c_{1,2}+c_{2,5}=1.41+2.24+1.00+1.41$	6.06
([5,2],[4,4])	$c_{t_2,t_6}+\theta_{6,6,6}$	$c_{5,4}+0=2.00+0$	2.00
([4,4],0)	$c_{t_6,0}$	$c_{4,0}$	2.24
Total			10.30

of the shortest path is 10.30 this is because it is necessary to add the cost of state $[i,i]=[2,2]$, that represents the departure from the main depot, in the recursion this is the first state with $F_{2,2}=c_{0,t_2}=c_{0,5}=1.00$. Then, $c(R_{2,6})=11.30$.

References

[1] Beasley JE. Route-first cluster-second methods for vehicle routing. Omega 1983;11:403–8.
[2] Bixby R. Solving real-world linear programs: a decade and more of progress. Operations Research 2002;50(1):3–15.
[3] Boudia M, Louly M, Prins C. A reactive GRASP and path relinking for a combined production-distribution problem. Computers & Operations Research 2007;34(11): 3402–19.
[4] Campos V, Laguna M, Martí R. Context-independent scatter and tabu search for permutation problems. INFORMS Journal on Computing 2005;17(1):111–22.
[5] Caramia M, Guerriero F. A heuristic approach for the truck and trailer routing problem. Journal of the Operational Research Society 2010;61:1168–80.
[6] Caramia M, Guerriero F. A milk collection problem with incompatibility constraints. Interfaces 2010;40(2):130–43.
[7] Chao I-M. A tabu search method for the truck and trailer routing problem. Computers & Operations Research 2002;29:33–51.
[8] Conover W. Practical nonparametric statistics. Wiley; 1998.
[9] Cordeau J-F, Gendreau M, Laporte G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks 1997;30:105–19.
[10] Desrosiers J, Soumis F, Desrochers M. Routing with time windows by column generation. Networks 1984;14(4):545–65.
[11] Dongarra JJ. Performance of various computers using standard linear equations software. Technical Report CS89-95, University of Tennessee, Computer Science; 2009.
[12] Dongarra JJ, Wade R, McMahan P. Linpack benchmark—Java version. 〈http://www.netlib.org/benchmark/linpackjava/〉; 2010.
[13] Drexl M. On some generalized routing problems. PhD thesis, Faculty of Business and Economics, RWTH Aachen University; 2007.
[14] Duhamel C, Lacomme P, Prins C, Prodhon C. A GRASP X ELS approach for the capacitated location-routing problem. Computers & Operations Research 2010;37(11):1912–23.
[15] Ehrgott M. Multicriteria optimization. Springer; 2000.
[16] El-Fallahi A, Prins C, Wolfler-Calvo R. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Computers & Operations Research 2008;35:1725–41.
[17] Feo TA, Resende MGC. Greedy randomized adaptive search procedures. Journal of Global Optimization 1995;6(2):109–33.
[18] Flood MM. The traveling-salesman problem. Operations Research 1956;4(1): 61–75.
[19] García S, Molina D, Lozano M, Herrera F. A study on the use of non-parametric tests for analyzing the evolutionary algorithms's behaviour: a case study on the CEC' 2005 special session on real parameter optimization. Journal of Heuristics 2009;15:617–44.
[20] Gendreau M, Potvin J-Y, Bräysy O, Hasle G, Løkketangen A. Metaheuristics for the vehicle routing problem and its extensions: a categorized bibliography. In: Golden B, Raghavan S, Wasil E, editors. The vehicle routing problem: latest advances and new challenges. Springer; 2008. p. 143–70.
[21] Gerdessen JC. Vehicle routing problem with trailers. European Journal of Operational Research 1996;93:135–47.
[22] Gillett BE, Miller LR. A heuristic algorithm for the vehicle-dispatch problem. Operations Research 1974;22(2):340–9.
[23] Glover F, Laguna M. Tabu search. Kluwer; 1997.
[24] Golden BL, Raghavan S, Wasil E. The Vehicle routing problem: latest advances and new challenges. Springer; 2008.
[25] Hansen P, Mladenovic N. Variable neighbourhood search: principles and applications. European Journal of Operational Research 2001;130:449–67.
[26] Hashimoto H, Yagiura M. A path relinking approach with an adaptive mechanism to control parameters for the vehicle routing problem with time windows. In van Hemert JI, Cotta C, editors, EvoCOP, Lecture notes in computer science, vol. 4972. Springer; 2008. p 254–65.
[27] Ho SC, Gendreau M. Path relinking for the vehicle routing problem. Journal of Heuristics 2006;12:55–72.
[28] Hoff A, Løkketangen A. A tabu search approach for milk collection in western Norway using trucks and trailers. In TRISTAN VI: The sixth triennial symposium on transportation analysis. Phuket, Tailand; June 10–15, 2007.
[29] Laguna M, Marti R. GRASP and path relinking for 2-layer straight line crossing minimization. INFORMS Journal on Computing 1999;11(1):44–52.

[30] Laporte G. What you should know about the vehicle routing problem. Naval Research Logistics 2007;54:811–9.

[31] Lin S-W, Yu VF, Chou S-Y. A note on the truck and trailer routing problem. Expert Systems with Applications 2010;37(1):899–903.

[32] Lin S-W, Yu VF, Chou S-Y. Solving the truck and trailer routing problem based on a simulated annealing heuristic. Computers & Operations Research 2009;36(5):1683–92.

[33] Or I. Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking. PhD thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL; 1976.

[34] Parragh SN, Doerner KF, Hartl RF, Gandibleux X. A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. Networks 2009;54(4): 227–42.

[35] Prins C. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research 2004;31:1985–2002.

[36] Prins C. A GRASP X evolutionary local search hybrid for the vehicle routing problem. In: Pereira FB, Tavares J, editors. Bio-inspired algorithms for the vehicle routing problem. Springer; 2009. p. 35–53.

[37] Prins C. Two memetic algorithms for heterogeneous fleet vehicle routing problems. Engineering Applications of Artificial Intelligence 2009;22:916–28.

[38] Prins C, Prodhon C, Wolfler-Calvo R. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR—A Quarterly Journal of Operations Research 2006;4:221–38.

[39] Reghioui M, Prins C, Labadi N. GRASP with path relinking for the capacitated arc routing problem with time windows. In: Giacobini M, Brabazon A, Cagnoni S, Caro GD, Drechsler R, Farooq M, Fink A, Lutton E, Machado P, Minner S, O'Neill M, Romero J, Rothlauf F, Squillero G, Takagi H, Uyar S, Yang S, editors. EvoWorkshops, Lecture notes in computer science, vol. 4448. Springer; 2007. p. 722–31.

[40] Resende MGC, Martí R, Gallego M, Duarte A. GRASP and path relinking for the maxmin diversity problem. Computers & Operations Research 2010;37(3): 498–508.

[41] Resende MGC, Ribeiro CC. GRASP with path-relinking: recent advances and applications. In: Ibaraki T, Nonobe K, Yagiura M, editors. Metaheuristics: progress as real problem solvers. Springer; 2005. p. 29–63.

[42] Resende MGC, Ribeiro CC. GRASP. AT&T Labs Research Technical Report, 2008.

[43] Resende MGC, Ribeiro CC. Greedy randomized adaptive search procedures: advances and applications. In: Gendreau M, Potvin J-Y, editors, Handbook of metaheuristics, 2nd ed., Springer; 2010, p. 281–317.

[44] Resende MGC, Werneck RF. A hybrid heuristic for the p-median problem. Journal of Heuristics 2004;10:59–88.

[45] Scheuerer S. A tabu search heuristic for the truck and trailer routing problem. Computers & Operations Research 2006;33:894–909.

[46] Schiavinotto T, Stutzle T. A review of metrics on permutations for search landscape analysis. Computers & Operations Research 2007;34(10):3143–53.

[47] Semet F. A two-phase algorithm for the partial accessibility constrained vehicle routing problem. Annals of Operations Research 1995;61:45–65.

[48] Semet F, Taillard E. Solving real-life vehicle routing problems efficiently using tabu search. Annals of Operations Research 1993;41:469–88.

[49] Sörensen K. Distance measures based on the edit distance for permutation-type representations. Journal of Heuristics 2006;13(1):35–47.

[50] Sörensen K, Sevaux M. A practical approach for robust and flexible vehicle routing using metaheuristics and Monte Carlo sampling. Journal of Mathematical Modelling and Algorithms 2009;8(4):387–407.

[51] Sörensen K, Sevaux M. Distance-based path relinking for the vehicle routing problem. In: TRISTAN VII: seventh triennial symposium on transportation analysis. Tromsø, Norway, June 20–25, 2010. p. 744–7.

[52] Souffriau W, Vansteenwegen P, Vanden Berghe G, VanOudheusden D. A path relinking approach for the team orienteering problem. Computers & Operations Research 2010;37(11):1853–9.

[53] Toth P, Vigo D. The vehicle routing problem. Philadelphia: SIAM; 2002.

[54] Villegas JG, Prins C, Prodhon C, Medaglia AL, Velasco N. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. Engineering Applications of Artificial Intelligence 2010;23(5):780–94.