



HAL
open science

Studies on collaborative transportation planning among carriers

Yuan Li

► **To cite this version:**

Yuan Li. Studies on collaborative transportation planning among carriers. Infrastructures de transport. Université de Technologie de Troyes, 2017. English. NNT : 2017TROY0008 . tel-02419745

HAL Id: tel-02419745

<https://theses.hal.science/tel-02419745>

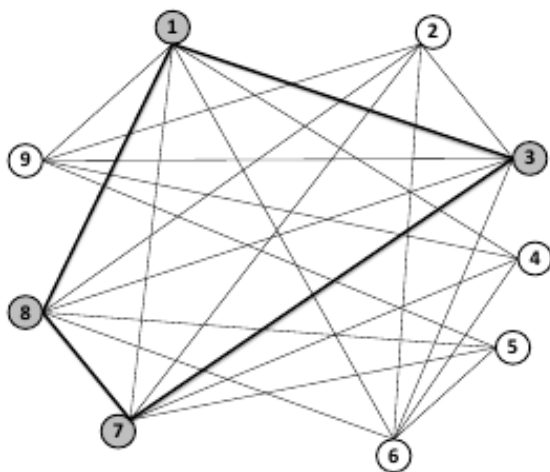
Submitted on 19 Dec 2019

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

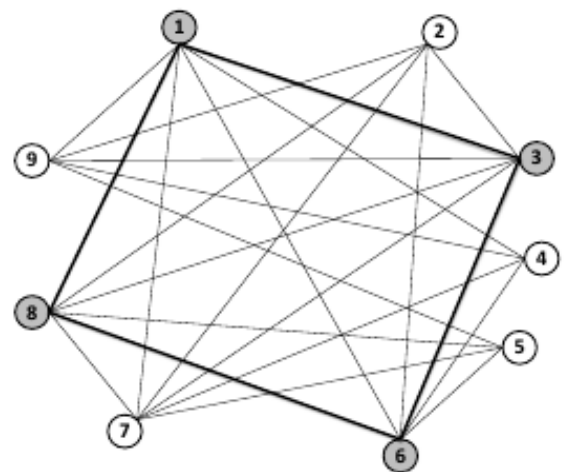
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Yuan LI

Studies on Collaborative Transportation Planning among Carriers



(a) Optimal clique for $RP(0)$



(b) Optimal clique for the original CE

Spécialité :
Optimisation et Sûreté des Systèmes

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Yuan LI

le 15 mars 2017

Studies on Collaborative Transportation Planning among Carriers

JURY

| | | |
|---------------------|-------------------------------|--------------------|
| M. R. WOLFLER CALVO | PROFESSEUR DES UNIVERSITES | Président |
| M. H. CHEN | PROFESSEUR DES UNIVERSITES | Directeur de thèse |
| M. D. FEILLET | PROFESSEUR ENSM SANTE ETIENNE | Rapporteur |
| M. H. KOPFER | PROFESSOR | Rapporteur |
| M. A. MOUKRIM | PROFESSEUR DES UNIVERSITES | Examineur |
| M. C. PRINS | PROFESSEUR DES UNIVERSITES | Directeur de thèse |

*To my Mother, my Father,
for their unlimited support, encouragement, and love.*

Acknowledgments

This work has been carried out within Laboratory of Industrial Systems Optimization (LOSI) at University of Technology of Troyes (UTT). It is funded by China Scholarship Council (CSC) and the project TCDU (Collaborative Transportation in Urban Distribution) of the ANR (French Research Agent). This project is labelled by 'Pôle Véhicule du Futur', and is jointly performed by four partners, three French Universities of Technology (UTT, UTBM, UTC) and a startup Share And Move Solutions (SAMS).

This work has been accomplished under the supervision of Prof. Haoxun Chen and Prof. Christian Prins. I would like to express my deepest gratitude to them for their highly valuable guidance and incessant support. I highly appreciate the friendly environment created during my three and a half year study. The high confidence they have given to me is the greatest reward of my endeavors.

I also would like to express my special thanks to M. Herbert KOPFER and M. Dominique FEILLET for accepting the review of my PhD thesis, and M. Aziz MOUKRIM and M. Roberto WOLFLER CALVO for agreeing to examine this thesis. Valuable remarks provided by the respectful experts in the field of collaborative logistics have helped me to improve the thesis' quality.

I would like to thank all researchers, secretaries and colleagues in LOSI, and the staffs of the doctoral school of UTT for their helps given to me in the past three and a half years.

I would like to thank all my friends, Dr. Feng TANG, Dr. Yubing PU, Dr. Tong QIAO, Dr. Aichun ZHU, Dr. Zhongmeng WEN, Dr. Wei GENG, Dr. Qingxian AN, M. Chao LU, for their helpful suggestions and encouragement during my stay in France.

Most of all, I would like to thank my parents for their supports, encouragements, and loves given to me in all my life.

Résumé

Dans la collaboration entre transporteurs, plusieurs transporteurs forment une alliance pour échanger leurs demandes de transport dans le but d'améliorer la rentabilité. Dans cette thèse, nous avons étudié la planification collaborative de transport entre transporteurs de charges partielles. Plus concrètement, nous avons étudié trois sous-problèmes soulevés dans cette planification collaborative : le problème de ramassage et de livraison avec fenêtres de temps, profits et demandes réservées, le problème de détermination de gagnants dans l'échange combinatoire, et le problème de génération d'enchère.

Ces trois sous-problèmes sont les problèmes clés pour la planification collaborative de transport parmi des transporteurs, et ils sont peu étudiés dans la littérature. Nous avons établi les nouveaux modèles de programmation mathématique pour ces problèmes et développé des heuristiques efficaces pour trouver des solutions très proches de leurs optimums dans un temps de calcul raisonnable. Les heuristiques proposées sont plus performantes que les solveurs commerciaux (GUROBI, CPLEX) non seulement en termes de la qualité de solution, mais aussi en termes du temps de calcul.

Mot clés : logistique collaborative, planification collaborative de transport, enchère combinatoire, méta-heuristique, problème de tournée de véhicule, optimisation combinatoire, programmation mathématique.

Abstract

In carrier collaboration, multiple carriers form an alliance to exchange their delivery requests for the purpose of improving their profitability. In this thesis, we have studied the collaborative transportation planning (CTP) among less-than-truckload (LTL) carriers. More concretely, we have studied three sub-problems raised in this collaborative planning : the pickup and delivery problem with time windows, profits, and reserved requests (PDPTWPR), the winner determination problem (WDP) in carrier collaboration via combinatorial exchange (CE), and the bid generation problem (BGP).

These sub-problems are the key issues for collaborative transportation planning among carriers, and they are rarely studied in the literature. We have established new mathematical programming models for these problems and developed efficient heuristics to find solutions close to their optimums in a reasonable computation time. The heuristics proposed are more efficient than commercial solvers (GUROBI, CPLEX) not only in terms of solution quality, but also in terms of computation time.

key words : collaborative logistics, collaborative transportation planning, combinatorial auctions, meta-heuristic, vehicle routing problem, combinatorial optimization, mathematical programming.

Contents

| | | |
|----------|--|-----------|
| 1 | General Introduction | 1 |
| 1.1 | Research background | 1 |
| 1.2 | Key issues in carrier collaboration | 2 |
| 1.3 | Problems studied in this thesis | 3 |
| 1.4 | Organisation of this thesis | 5 |
| 2 | Literature review of CTP | 7 |
| 2.1 | Introduction | 7 |
| 2.2 | Collaborative logistics | 10 |
| 2.3 | Centralized planning approaches for collaborative transportation | 12 |
| 2.4 | Decentralized planning approaches for CL | 13 |
| 2.4.1 | Auction-based mechanisms | 14 |
| 2.4.2 | Bid generation problems | 20 |
| 2.5 | Conclusion | 22 |
| 3 | PDP with time windows, profits and reserved requests | 23 |
| 3.1 | Introduction | 24 |
| 3.2 | Literature review | 26 |
| 3.3 | Problem description and mathematical model | 28 |
| 3.4 | Adaptive large neighborhood search | 31 |
| 3.4.1 | Initial solution construction | 34 |
| 3.4.2 | Adaptive selection of destroy/repair operators | 36 |
| 3.4.3 | Dynamic adjustment of operator behavior | 38 |

| | | |
|----------|---|-----------|
| 3.4.4 | Description of destroy/repair operators | 40 |
| 3.4.5 | Diversification via simulated annealing | 45 |
| 3.4.6 | Local search procedure | 45 |
| 3.5 | Computational experiments | 48 |
| 3.5.1 | Generation of instances | 49 |
| 3.5.2 | Parameter setting | 50 |
| 3.5.3 | Experimental results | 51 |
| 3.5.4 | Impact of the DAOB mechanism | 55 |
| 3.6 | Conclusion | 56 |
| 4 | Stochastic bid generation problem in LTL transportation | 59 |
| 4.1 | Introduction | 60 |
| 4.2 | Problem description and mathematical model | 62 |
| 4.2.1 | The stochastic MIQP model and its linearization | 68 |
| 4.2.2 | Deterministic transformation of the probabilistic constraint | 69 |
| 4.2.3 | Equivalent MIP model | 70 |
| 4.2.4 | Synergy factor estimation | 72 |
| 4.3 | GRASP \times ILS hybrid | 75 |
| 4.3.1 | Initial solution construction | 77 |
| 4.3.2 | Local search operators | 77 |
| 4.3.3 | Mutation | 79 |
| 4.4 | Computational experiments | 79 |
| 4.4.1 | Generation of instances | 79 |
| 4.4.2 | Parameter setting | 81 |
| 4.4.3 | Experimental results | 81 |

| | |
|--|------------|
| Contents | ix |
| <hr/> | |
| 4.5 Conclusions | 84 |
| 5 WDP in Carrier Collaboration Via CE | 87 |
| 5.1 Introduction | 88 |
| 5.2 Literature review | 90 |
| 5.2.1 Winner determination problems | 90 |
| 5.2.2 Maximum vertex weight clique problems | 92 |
| 5.3 Combinatorial exchange versus combinatorial auction | 93 |
| 5.3.1 Better exploration of synergies among requests | 94 |
| 5.3.2 More efficient allocation due to a larger solution space | 95 |
| 5.4 Problem description and mathematical model | 97 |
| 5.4.1 Formulation of the winner determination problem of the CE | 98 |
| 5.4.2 Bidding languages of combinatorial exchange | 99 |
| 5.5 Lagrangian relaxation approach | 100 |
| 5.5.1 Framework of Lagrangian relaxation approach | 100 |
| 5.5.2 Solving the Lagrangian relaxed problem | 104 |
| 5.5.3 Update of Lagrangian multipliers | 109 |
| 5.5.4 Repair heuristic | 109 |
| 5.6 Computational experiments | 111 |
| 5.6.1 Generation of instances | 112 |
| 5.6.2 Parameter setting | 115 |
| 5.6.3 Experimental results | 115 |
| 5.7 Conclusions | 118 |
| 6 Conclusions and Perspectives | 121 |

| | | |
|----------|--|------------|
| 6.1 | Conclusions | 121 |
| 6.2 | Perspectives | 123 |
| A | Appendix of French Abstract : | 127 |
| A.1 | Introduction générale | 127 |
| A.2 | Introduction | 131 |
| A.3 | Le problème de ramassage et de livraison avec fenêtres de temps, profits et demandes réservées | 134 |
| A.4 | Le problème de génération d'enchère dans le transport de chargement partial | 140 |
| A.5 | Le problème de détermination de gagnants dans la collabo- ration entre transporteurs via l'échange combinatoire | 143 |
| A.6 | Conclusion et perspective | 146 |
| | References | 149 |

List of figures

| | | |
|-----|--|-----|
| 3.1 | Illustration of neighborhoods in a classical ALNS and in our DAOB version. | 40 |
| 3.2 | Intraroute relocate : delivery node of request A is relocated . | 47 |
| 3.3 | Interoute relocate : request B is relocated | 47 |
| 3.4 | Intraroute exchange : request A and request C are exchanged | 47 |
| 3.5 | Interoute exchange : request B and request D are exchanged | 47 |
| 3.6 | Selective requests removal : request B is removed | 48 |
| 3.7 | Selective requests insertion : request B is inserted | 48 |
| 3.8 | Proportion of feasible solutions generated with or without DAOB | 56 |
| 4.1 | Example of cost calculation for synergy factor estimation for carrier <i>a</i> | 74 |
| 4.2 | Example of cost calculation for synergy factor estimation for carrier <i>b</i> | 74 |
| 4.3 | Example of cost calculation for synergy factor estimation for carrier <i>c</i> | 75 |
| 5.1 | Comparison of routing plannings among no-collaboration scenario, CA and CE. | 97 |
| 5.2 | Comparison of optimal cliques between the relaxed problem and the CE problem. | 106 |

List of tables

| | | |
|-----|---|----|
| 3.1 | Adaptive adjustment of the operator scores | 37 |
| 3.2 | Parameter tuning according to instance size | 51 |
| 3.3 | Abbreviation of experiment indicators and definition | 52 |
| 3.4 | Computational results for small size instances | 54 |
| 3.5 | Computational results for medium size instances | 54 |
| 3.6 | Computational results for large instances test | 55 |
| 3.7 | Statistical average value on the performance indicators | 55 |
| 3.8 | Average improvement of total profit with the DAOB | 56 |
| 4.1 | Example data for synergy factor estimation | 74 |
| 4.2 | Parameter setting of the GRASP \times ILS according to instance size | 81 |
| 4.3 | GRASP \times ILS versus GUROBI MIP solver on small size instances ($\mu = \rho \times distance \times load$) | 82 |
| 4.4 | GRASP \times ILS versus GUROBI MIP solver on small size instances ($\mu = \rho \times distance$) | 83 |
| 4.5 | GRASP \times ILS versus GUROBI MIP solver on medium size instances ($\mu = \rho \times distance \times load$) | 83 |
| 4.6 | GRASP \times ILS versus GUROBI MIP solver on medium size instances ($\mu = \rho \times distance$) | 84 |
| 5.1 | Data for the illustrative example | 96 |
| 5.2 | Comparison of results among no-collaboration scenario, CA and CE | 97 |

| | | |
|-----|--|-----|
| 5.3 | Bids information of the example | 106 |
| 5.4 | Parameter setting according to instance size | 115 |
| 5.5 | Lagrangian relaxation versus MIP on small size instances . . | 116 |
| 5.6 | Lagrangian relaxation versus MIP on medium size instances | 117 |
| 5.7 | Lagrangian relaxation versus MIP on large size instances . . | 118 |

List of Abbreviations

| Acronym | Meanings |
|----------------|--|
| ALNS | Adaptive Large Neighborhood Search |
| BGP | Bid Generation Problem |
| BPO | Bid Price Optimization |
| CA | Combinatorial Auction |
| CE | Combinatorial Exchange |
| CL | Collaborative Logistics |
| CTP | Collaborative Transportation Planning |
| CTM | Collaborative Transportation Management |
| LB | Lower Bound |
| ILS | Iterative Local Search |
| IP | Integer Programming |
| LR | Lagrangian Relaxation |
| LS | Local Search |
| LTL | Less-Than-Truckload |
| MIP | Mixed Integer Programming |
| MIQP | Mixed Integer Quadratic Programming |
| MNTS | Multi-Neighborhood Tabu Search |
| MVWCP | Maximum Vertex Weight Clique Problem |
| PDPTWPR | Pickup and Delivery Problem with Time Windows, Profits, and Reserved requests |
| SA | Simulated Annealing |
| TL | Truckload |
| UB | Upper Bound |

| | |
|--------|---|
| VNS | Variable Neighborhood Search |
| VRP | Vehicle Routing Problem |
| VRPP | Vehicle Routing Problem with Profits |
| VRPTW | Vehicle Routing Problem with Time Windows |
| WDP | Winner Determination Problem |
| 0-1 IP | 0-1 Integer Programming |

Glossary of Notations

| Notation | Definition |
|---------------------|---|
| x_{ij}^k | Binary variable equals to 1 if and only if vehicle k travels directly through arc (i, j) . |
| y_i^k | Binary variable equals to 1 if and only if request i is served by vehicle k . |
| T_i^k | Time at which vehicle k begins service at node i . |
| Q_i^k | Load of vehicle k when leaving node i . |
| z^b | Binary variable equals to 1 if and only if bundle b is a generated bid. |
| p^b | The 'ask price' of bundle b . |
| R_r | Set of carrier's reserved requests. |
| R_s | Set of selective requests or requests for auction. |
| R | Set of all requests, $R = R_r \cup R_s$. |
| B | Set of bids, indexed by b . |
| K | Set of available vehicles, indexed by k . |
| e_i^b | Binary parameter equals to 1 if and only if request i is in bid b , otherwise 0. |
| p_{min}^b | Small positive value indicating the minimum 'ask price' of bid b . |
| Y^b | Random variable denoting the lowest price (auction clearing price) offered by competitors for bid b . |
| $\alpha \in [0, 1]$ | Probability threshold. |
| $p(R_r)$ | Total revenue obtained from serving the reserved requests. |

| | |
|----------------------|---|
| p_i | Revenue of request i . |
| c_{ij} | Transportation cost from node i to j , where $c_{ij} = c_{ji}$, and the triangle inequality holds. |
| t_{ij} | Travel time from node i to j . |
| d_i | Quantity of request i . |
| $[a_i, b_i]$ | Time windows of node i . |
| Q | Vehicle capacity. |
| $T_{ij} = b_j - a_i$ | Big-M constant in the time window constraints. |
| $Q_i = Q + d_i$ | Big-M constant in the capacity constraints. |
| N | Set of carriers in a coalition. |
| M | Set of requests that carriers wish to exchange. |
| B | Set of all bids. |
| p_j | The price of bid j . |
| $a_{ik} = 1$ | Binary parameter equals to 1 if and only if request k can be sold out by carrier i . |
| $b_{ik} = 1$ | Binary parameter equals to 1 if and only if request k can be bought in by carrier i . |
| x_j | Binary variable equals to 1 if and only if bid j is a winning bid. |

General Introduction

1.1 Research background

The collaboration among small or medium sized enterprises (SME) plays a growing role in their daily operation/management. Participation in a network and collaboration with other enterprises has now become the primary strategy for them to gain competitive advantages in current severe environment. To achieve economies of scale, more and more SMEs have formed collaborative networks by sharing tasks and resources, in order to reduce costs, improve responsiveness to the evolution of market demands, and capture more business opportunities.

Fierce competition in global markets, introduction of products with shorter life cycles, increasing fuel costs and labor prices, growing transportation legislation and heightened expectations of customers have shrunk profit margins of carriers (Cruijssen, Cools, & Dullaert, 2007). Thus, as an effective strategy for small to medium-sized freight carriers to improve profitability by reducing empty vehicle repositions and increasing vehicle fill rates, carrier collaboration is emerging and attracting a growing interest from industrial practitioners and academic researchers (Dai & Chen, 2009). Some pilot projects implemented in USA reveal that Collaborative Transportation Management (CTM) (including carrier collaboration) can reduce

the mileage traveled by empty vehicles by 15%, the waiting and pause time of vehicles by 15%, the turnover of drivers by 15% and can increase the fill rate of vehicles by 33% (Sutherland, 2009).

Carrier collaboration refers to the partnership among multiple carriers at the same level of logistics operations. Multiple carriers form a coalition and exchange their transportation requests to explore better complementarity among requests. Better complementarity among requests can improve the routing planning by increasing the vehicle fill rates or eliminating empty backhauls of carriers and consequently reduce their transportation costs. One problem for carrier collaboration is how to optimally exchange (reallocate) requests among carriers so that their total profit is maximized. The other problem is how to fairly allocate the post-collaboration profit gained through the collaboration among all carriers in the coalition in order to guarantee the sustainability of the alliance (Chen, 2016). In this thesis, we focus on the first problem which is also referred to as collaborative transportation planning (CTP) problem (Wang & Kopfer, 2011, 2014; Wang, Kopfer, & Gendreau, 2014).

1.2 Key issues in carrier collaboration

In this thesis, we consider carrier collaboration realized through a combinatorial auction or exchange. Such collaboration mechanism involves a virtual/real auctioneer and multiple carriers. Each carrier plays a double role of buyer and seller of transportation requests for the purpose of improving its profitability through exchanging its requests with other carriers. The auctioneer is responsible for solving a winner determination problem (WDP) (Ackermann, Ewe, Kopfer, & Küfer, 2011) to reallocate requests to

the winning carriers. The following sequence of events describes the procedure of requests exchange in less-than-truckload (LTL) transportation for carrier collaboration :

1. Each carrier evaluates its requests as profitable or unprofitable by solving the pickup and delivery problem with time windows, profits and reserved requests (PDPTWPR) (Chapter 3).
2. Unprofitable requests are submitted to the auctioneer as outsourcing requests.
3. The auctioneer announces the requests for auction to all carriers.
4. Each carrier tenders bid(s) with an 'ask price' to the auctioneer, and each bid is composed of one or more requests. This problem is referred to as the bid generation problem (BGP) (Triki, Oprea, Beraldi, & Crainic, 2014; Buer, 2014; Kuyzu, Akyol, Ergun, & Savelsbergh, 2015) (Chapter 4).
5. The auctioneer solves the WDP to reallocate requests among carriers according to the winning bids (Chapter 5).

1.3 Problems studied in this thesis

The goal of this thesis is to investigate the above mentioned sub-problems of carrier collaboration. More specifically, this thesis mainly studied the sub-problems appeared in steps 1,4,5 mentioned above. Thus, this thesis is devoted to solve the following three problems :

1. **The pickup and delivery problem with time windows, profits, and reserved requests (PDPTWPR)**, a new vehicle routing problem appeared in combinatorial auction (CA) for less-than-truckload (LTL)

transportation. This problem is also the first step of the proposed carrier collaboration framework in this thesis, for each carrier to identify the set of profitable requests and the set of unprofitable requests. A mixed-integer programming (MIP) model is formulated for the PDPTWPR and an adaptive large neighborhood search (ALNS) approach is developed as the solution approach. The ALNS involves ad-hoc destroy/repair operators and a local search procedure. It runs in successive segments which change the behavior of operators and compute their own statistics to adapt selection probabilities of operators. The MIP model and the ALNS approach are evaluated on 54 randomly generated instances with up to 10-100 requests. The numerical results indicate that the ALNS significantly outperforms the commercial solver, not only in terms of solution quality but also in terms of CPU time.

2. **The stochastic bid generation problem (BGP)** appears in the bidding phase of our carrier collaboration framework. The stochastic BGP combines request selection problem and routing problem for serving pickup and delivery requests with time windows. A mixed-integer quadratic programming (MIQP) with a probabilistic constraint is formulated for the problem. To tackle the intractable MIQP model, deterministic transformation and linearization technique are used to transform it into a MIP model. The MIP model can be decomposed into several independent sub-problems, and each of them is a the pickup and delivery problem with time windows (PDPTW). A multi-start iterative local search (GRASP \times ILS) algorithm is then developed to solve the simplified model. Extensive numerical experiments comparing the algorithm with the GUROBI

MIP solver on randomly generated instances show the effectiveness of the algorithm.

3. **Winner determination problem (WDP) in carrier collaboration via combinatorial exchange** aims to reallocate transportation requests among carriers, which is the last step of the carrier collaboration framework in this thesis. The CE can be considered as an alternative approach of CA in LTL transportation. But in the CE, each carrier plays a double role of buyer and seller of transportation requests. This characteristic brings some advantages compared with traditional CAs in freight logistics. A 0-1 linear programming model is formulated for this problem. A Lagrangian relaxation approach is developed to solve the CE problem. The relaxed problem is transformed into a maximum vertex weight clique problem (MVWCP) which is solved either by a multi-neighborhood tabu search or by a commercial MIP solver. A repair heuristic is proposed to cope with any infeasibilities caused by the constraint relaxation. Extensive numerical experiments on randomly generated instances show that the Lagrangian relaxation approach can provide high quality solutions.

1.4 Organisation of this thesis

Chapter 1 mainly introduces our research background and generally describes the problems studied in this thesis. Chapter 2 provides an overview of current-art of collaborative transportation planning. An general review is given firstly to the field of collaborative logistics. Then the literature review focuses on significant contributions and important review papers on the decentralized planning approaches, especially the auction-

based mechanisms, for carrier collaboration. Chapter 3 is devoted to solve the PDPTWPR. Chapter 4 focuses on the stochastic BGP. Chapter 5 solves the WDP in carrier collaboration via combinatorial exchange. At last, Chapter 6 concludes this thesis and present the perspectives of future works opened in the field of collaborative transportation planning.

Literature review of Collaborative Transportation Planning

Contents

| | |
|---|-----------|
| 2.1 Introduction | 7 |
| 2.2 Collaborative logistics | 10 |
| 2.3 Centralized planning approaches for collaborative trans- portation | 12 |
| 2.4 Decentralized planning approaches for CL | 13 |
| 2.4.1 Auction-based mechanisms | 14 |
| 2.4.2 Bid generation problems | 20 |
| 2.5 Conclusion | 22 |

2.1 Introduction

Freight logistics specializes in the movement (or 'forwarding') of freight, or cargo, from one place to another. In the last decade, with the flourishment of E-commerce and economic globalization, freight forwarding business has been playing an essential role in daily economic activities. However, the rapid development of freight logistics induces a fierce competi-

tion among freight carriers or forwarders. Moreover, introduction of products with shorter life cycles, increasing fuel costs and labor prices, growing transportation legislation and heightened expectations of customers have shrunk profit margins of carriers (Crujssen et al., 2007). For small or medium sized carriers, how to survive in such unprecedented competition environments poses a real challenge. The challenge has given rise to Collaborative Logistics (CL) or Collaborative Transportation Management (CTM). CL or CTM is achieved through the horizontal collaboration between multiple shippers or carriers by either sharing transport capacities or transportation orders. With the collaboration, all actors involved can improve their profitability by eliminating empty backhauls and raising vehicle utilization rates (Dai, 2011). Note that such collaboration benefits from the development of information technology in recent years.

Horizontal collaborative logistics refers to the collaboration among multiple actors at the same level in logistics operations such as the collaboration among shippers (manufacturers) and the collaboration among carriers. Two types of horizontal collaborative logistics are studied in the literature : shipper collaboration and carrier collaboration. Shipper collaboration (Ö. Ergun, Kuyzu, & Savelsbergh, 2007) considers the situation of a single carrier and multiple shippers. The collaboration among shippers is realized by consolidation of their transportation requests to be offered to carriers. Through collaboration, shippers are able to reduce 'hidden costs' such as asset reposition costs. However, more attention has been given to carrier collaboration. Differing from shipper collaboration, carrier collaboration (Özener, Ergun, & Savelsbergh, 2011 ; Hernández, Unnikrishnan, & Awale, 2012) happens among multiple carriers and considers how to provide opportunities for carriers to exploit synergies among their transpor-

tation orders (requests) in daily operations, reduce costs associated with fleet operation, decrease lead times, increase asset utilization, and enhance overall service levels (Esper & Williams, 2003).

In realistic logistics services, two types of transportation services are often provided to customers : truckload (TL) transportation and less-than-truckload (LTL) transportation. TL shipping refers to the movement of large amounts of homogeneous cargoes from one origin to one destination, whereas LTL refers to the shipping of relatively small freights from multiple origins to multiple destinations. For LTL shipment, customers pay for the amount of space of capacity used on the truck, instead of the entire truck in TL transportation. This advantage makes one customer share the space of a truck with other customers in case of shipping small freights. This option is ideal for small to medium sized enterprises who do not have their own trucks or can not afford TL shipping. Note that 'truck' used in this thesis may refer to other transportation tools, such as vans or vehicles. These terms are exchangeable hereafter.

One problem for carrier collaboration is how to optimally exchange (reallocate) requests among carriers so that their total profit is maximized, which is also referred to as collaborative transportation planning (CTP) problem (Wang & Kopfer, 2011, 2014; Wang et al., 2014). The other problem is how to fairly allocate the post-collaboration profit gained through the collaboration among all carriers in the coalition in order to guarantee the sustainability of the alliance. The profit allocation will not be discussed in the thesis, we refer readers to a recent review paper on this issue (Guajardo & Rönnqvist, 2016).

In this thesis, we focus on the CTP for carrier collaboration in LTL mode. Two types of approaches for such problem can be found in the literature :

centralized planning approaches and decentralized planning approaches. In centralized planning approaches, a central coordinator is responsible for optimally reassigning transportation orders among carriers so that their total profit is maximized. On the contrary, in decentralized planning approaches, each carrier acts as an autonomous agent, there is no central coordinator to arrange the overall plan of all carriers involved. Because of the nature of centralized planning approaches, all transportation orders' information is open to each of other carriers in coalition, whereas only limited information of orders is shared among carriers in decentralized planning approaches. Although centralized methods are often superior to decentralized methods in terms of total profit or cost, decentralized approaches are more practical in realistic applications. Because carriers are not only partners but also competitors, they may be not willing to disclose customers' orders and cost information even to their cooperation partners (Verdonck, Caris, Ramaekers, & Janssens, 2013).

The rest of this chapter will be organized as follows : a general literature review on collaborative logistics will be given firstly in Section 2.2. Then a brief review on centralized planning approaches and a detailed review on decentralized planning approaches will be given in Section 2.3 and Section 2.4 separately, since the three sub-problems studied in this thesis are only related to the class of decentralized planning approaches.

2.2 Collaborative logistics

Crujssen et al. (2007) launch a large-scale survey on the opportunities and impediments of horizontal collaboration in logistics. In general, logistics service providers (LSPs) strongly believe that the potential benefits of

horizontal collaboration can increase their profitability or improve the quality of their services. However, the impediments for collaboration that are perceived or expected by the non-cooperating LSPs have been proved to be experienced by the cooperating LSPs. They believe a fair allocation of profit is the hardest challenge.

Kopfer et Krajewska (2007) provide an overview and a comparison of existing approaches for modeling and solving the integrated transportation and forwarding problem, which is an extended problem of the traditional routing and scheduling problem in freight forwarding provided by 3rd and 4th party logistics. This study summarizes the reasons for the existence of the gap between theory and practice.

D'Amours et Rönnqvist (2010) present a survey of previous contributions in the field of collaborative logistics. Firstly, they depict opportunities in collaborative transportation planning. Then they discuss key issues in forming coalitions, such as sharing resources and profits, as well as the issues about information protection and decisions technologies. Some business cases are also given to support the discussion in the paper. Finally, they raise some question and describe perspectives of future research.

Verdonck et al. (2013) provide a thorough literature review on the operational planning related to horizontal logistics collaboration. In this review paper, the authors classify the horizontal logistics collaboration into two mainstream approaches : order sharing and capacity sharing. For both research streams, a detailed overview of solution techniques is presented. For order sharing approaches, carriers may achieve an increase in capacity utilization, improved asset repositioning capabilities and a reduction in total transportation costs due to enhanced transportation planning. Existing studies address distinct approaches to tackle order sharing by optimal re-

allocation of requests. The authors classify order sharing approaches into five types : joint route planning, auction-based mechanisms, bilateral lane exchanges, load swapping and shipment dispatching policies. Instead of sharing customer orders, carriers may also collaborate with each other horizontally through the sharing of vehicle capacities. In this way, capital investments may be shared among partners and utilization rates of vehicles may be improved. Previous studies provide two general techniques to determine the most efficient way to share vehicle capacities, namely the way based on mathematical programming and the way based on negotiation protocols. At last, some promising future research directions are proposed in the field of collaborative logistics.

2.3 Centralized planning approaches for collaborative transportation

For centralized planning approaches in collaborative logistics with TL transportation, [O. Ergun, Kuyzu, et Savelsbergh \(2003\)](#) propose a Lane Covering Problem (LCP), which is a centralized planning model and arises in the context of shipper collaboration. The object function of the model aims to find a minimum cost set of directed cycles (truck tours), not necessarily disjoint, covering a given subset of arcs (lanes) in a complete Euclidean digraph. Effective algorithms and efficient implementations for solving the LCP are also developed in this paper. In later studies, the authors consider some constrained variants of LCP, such as the cardinality constrained LCP ([Ö. Ergun et al., 2007](#)), the length constrained LCP ([O. Ergun et al., 2003](#)), the dispatch windows constrained LCP and driver restriction LCP ([O. Er-](#)

gun, Kuyzu, & Savelsbergh, 2007b). Ozener, Ergun, et Savelsbergh (2009) investigate the potential of collaborative opportunities among carriers in TL transportation. In the proposed optimization models, carriers' repositioning costs may be reduced by exchanging transportation requests. Various exchange mechanisms are designed and evaluated, differing in terms of information sharing requirements and side payment options, which allow carriers to realize potential costs savings opportunities.

For centralized planning approaches in collaborative logistics with LTL transportation, Hernández et Peeta (2014) study a single-carrier collaboration problem (SCCP) in LTL transportation, in which a carrier seeks to collaborate with other carriers in order to acquire extra transport capacity to service excess demand. The SCCP is considered from a static planning perspective to gain insights into the potential benefits of the collaboration concept for carriers, and its ability to mitigate the consumption of fuel. The collaborative strategies are evaluated by computing the relative benefits of the collaboration over the non-collaboration situation. Single and multiple-product SCCPs are both formulated as binary (0-1) multi-commodity minimum cost flow problems, and the models are solved by a branch-and-cut algorithm.

2.4 Decentralized planning approaches for collaborative transportation

In CTP, decentralized planning approaches are achieved by auction-based mechanisms in most cases. Our study in chapter 5 also aims to tackle such problem by combinatorial auctions/exchanges, so we review it

thoroughly in Subsection 2.4.1.

The bid generation problem (BGP) is a key decision problem for auction-based decentralized planning approaches in CTP, which is considered from the perspective of each carrier. The BGP can also be considered as the request selection problem. Our study in chapter 3 and chapter 4 are related to the request selection problem, so the BGP will be reviewed in detail in Subsection 2.4.2.

2.4.1 Auction-based mechanisms

Auction-based approaches are the most important methods for CTP. An auction is a process of buying and selling goods or services by offering them up for bid, taking bids, and then selling the item to the winning bidder. Different types of auctions exist in the literature, such as single-object auctions, multiple-object auctions, combinatorial auctions (CA), and exchanges (Shoham & Leyton-Brown, 2008; Krishna, 2009).

As important category of auctions, combinatorial auctions allow bidders to bid on combinations of objects, tend to lead to more efficient allocations than traditional auction mechanisms in multi-object auctions where the agents' valuations of the objects are not additive. However, determining the winners of an auction so as to maximize the revenue of the auctioneer or the total profit of all bidders is NP-complete (Sandholm, 2002). Such problem is also referred to as the combinatorial auction problem (CAP) or the winner determination problem (WDP). Various combinatorial auction mechanisms are introduced in Cramton, Shoham, et Steinberg (2006), such as Vickrey-Clarke-Groves auctions, iterative combinatorial auctions, simultaneous ascending auctions, ascending proxy auctions, and clock-proxy auc-

tions.

Here, we mainly introduce two kinds of combinatorial auctions applied for CTP in the literature : single-round combinatorial auctions and multi-round combinatorial auctions (iterative combinatorial auctions). Note that multi-round combinatorial auctions have several advantages over the single-round combinatorial auctions (De Vries & Vohra, 2003; Kwon, Lee, & Ma, 2005). Their applications in CTP will be reviewed in the following subsections, respectively.

2.4.1.1 Single-round combinatorial auctions

Auctions are used in transportation planning early in Song et Regan (2003a). A framework of auction-based request allocation mechanism for carriers is proposed in this work. Their analysis shows their proposed auction-based system is Pareto efficient (Pareto, 1971; Pardalos, Migdalas, & Pitsoulis, 2008). Meanwhile, the related complex decision issues like sub-contracting, bid generation, bid selection are also investigated in this paper. In the same year, Song et Regan (2003b) study the complexity of the bidding problem in CA for the procurement of TL trucking service contracts. Furthermore, an optimization-based approximation method is developed to help a carrier to construct its bids.

Sheffi (2004) investigates a CA application in transportation service procurement from shippers' point of view. His work proves that the combinatorial bidding strategy allows both shippers and carriers to exploit inherent economies of scope in TL operations.

Figliozzi (2006) proposes collaborative mechanisms (CMs) for carrier collaborations. Three game theoretic properties : Budget Balanced (BB), In-

dividually Rational (IR) and Incentive Compatible (IC) (Babaioff & Walsh, 2005; Nisan, 2007) are necessary to guarantee sustainability and efficiency of the CMs. A second-price-based dynamic collaborative mechanism (SPDCM) is employed for the auction procedure. A simulation study is carried out on a hypothetical coalition of four carriers. Results clearly show that the proposed collaborative mechanism outperforms the non-collaborative situation.

Krajewska et Kopfer (2006) present a three-phase request allocation scheme based on CA and game theory. In the pre-processing phase, each request is estimated by 'potential self-fulfilment cost'. Then orders are exchanged via a modified matrix auction (Day, 2004; Goossens & Spijksma, 2007) in the profit optimization phase. Finally, the collaborative profit is shared based on a game theory concept in the profit sharing phase.

For CA applied to carrier collaboration in LTL transportation, Schwind, Gujo, et Vykoukal (2009) develop a combinatorial exchange mechanism ComEx system, which exchanges delivery orders in a logistics company organized by profit centers. The ComEx system has four phases : initialization phase, outsourcing phase, insourcing phase, and final evaluation phase. Outsourcing requests to other profit centers is determined in the outsourcing phase, and the acquiring requests from other profit centers is executed in the insourcing phase. In the final evaluation phase, CA is used to minimize the total cost to allocate delivery orders among profit centers. Simulation tests based on real data from a real-world medium-sized logistics company shows the ComEx can achieve up to 14% cost saving.

The issue about disclosing of business information among carriers is a key focus in Berger et Bierwirth (2010). Their work proposes a decentralized control and auction based exchange mechanism for the request reas-

signment problem in collaborative carrier networks. The decentralized approach, the centralized one and no-collaboration situation are compared in this paper. The simulation makes evident that the centralized approach yields the best outcome, but at a cost of information sharing. On the other hand, no-collaboration situation is dominated by the collaboration using decentralized approach. Note that they ignore the vehicle capacity by assuming that all shipments take only a very small fraction of the space of a vehicle.

[Ackermann et al. \(2011\)](#) discuss various objectives of a combinatorial request exchange mechanism in freight logistics and provide the arguments for their designed decisions. Their study is based on a realistic environment of LTL carriers. The proposed mechanism is flexible since it allows the unbundling of bundles into sub-bundles in the offering phase and regrouping them later in the auction phase.

A recent exploratory study on collaborative urban logistics in Singapore ([Handoko & Lau, 2016](#)) considers the collaboration among carriers through the exchange of shareable orders (exchangeable transportation requests) at an urban consolidation center for their last mile deliveries. A single-round sealed-bid double auction is proposed for the order exchange. In this mechanism, all bids with an ask price higher than the offer price are banned and a winner determination problem is solved to reallocate the exchangeable transportation requests among carriers.

2.4.1.2 Iterative combinatorial auctions

[Kwon et al. \(2005\)](#) consider an integrated multi-round combinatorial auction mechanism for TL transportation procurement. On one hand, ship-

pers allow bids on packages of lanes (requests) and solve a winner determination problem to assign lanes to carriers. On the other hand, carriers employ bid generation models to discover profitable lanes to bid for at each round. Price information derived from the result of a winner determination solution is used by carriers to determine profitable lanes for the next round of bidding. Hence, the mechanism proposed in this study integrates the optimization models of shippers and carriers. The numerical results show the benefits of the collaboration to two both sides.

Dai et Chen (2011) present a multi-agent and iterative auction-based framework for carrier collaboration in LTL with pickup and delivery requests. This approach is fully decentralized where each carrier plays both auctioneer and bidder role and no post-coordination profit reallocation is needed. Each carrier must asynchronously solve an outsourcing requests selection problem (ORSP) and a requests bidding problem (RBP), both problems are mathematically formulated. Any new-coming requests can be dynamically added into the request pool. Simulation tests on 20 randomly generated instances reveal their approach can achieve a profit increasing compared with a no collaboration situation.

Wang et Kopfer (2014) implement a route-based iterative combinatorial auction for CTP of LTL freight carriers. In each round of the auction, each carrier tenders its routing plan to the auctioneer, which is generated based on the dual values of the linear relaxation of a set partitioning model. Then, the auctioneer solves a provisional winner determination problem that minimizes the total fulfillment costs of all carriers. At the end of this auction, a final winner determination problem is solved to allocate the routes to the winning carriers. This approach assumes that all transportation requests of the carriers are offered for exchange and the ask price

of each route determined based on its fulfillment cost must be revealed to the auctioneer. Computational simulation reveals the route-based exchange mechanism can realize a cost saving up to 18%.

Dai, Chen, et Yang (2014) develop a multi-round auction based on a price-setting mechanism to achieve the collaboration among LTL carriers. Their approach neither need to disclose any confidential information nor require to solve a NP-hard winner determination problem. In this auction, a virtual auctioneer who plays the role of mediator between carriers. In each round, every carrier determines the requests to bid for based on the price of serving each outsourcing request announced by the auctioneer. The auctioneer will update the prices according to the bids of all carriers it receives. The auction procedure stops when a certain stopping criterion is met. Three different price adjustment methods are compared to prove the effectiveness of the approach.

J. Li, Rong, et Feng (2015) propose a multi-round auction for carrier collaboration in TL transportation with pickup and delivery requests. But their auction mechanism only allows single request exchange. In each round of the auction, each carrier announces one request to outsource and one request to insource based on two request selection models for outsourcing and insourcing, respectively. Although the restriction to one request to outsource and one request to source in each round simplifies the auction process and lighten the computational burden, it may obtain a near-optimal solution.

Recently, Chen (2016) extends the clock-proxy auction to a combinatorial clock-proxy exchange for a carrier collaboration problem in LTL transportation. This is also an iterative combinatorial auction, which has two phases. The first phase is called clock phase, where an iterative exchange

based on Lagrangian relaxation is developed. The second phase is called proxy phase, where the bids that each carrier submits to its proxy agent are determined based on the information observed in the clock phase. The proposed approach combines the simple and transparent price discovery of the clock exchange with the efficiency of the proxy exchange. Computational results on randomly generated instances show the usefulness of the proxy phase and the effectiveness of the clock-proxy exchange.

2.4.2 Bid generation problems

The BGP in carrier collaboration can be classified into two categories : deterministic BGPs and stochastic BGPs. We will review them separately in the following subsections.

2.4.2.1 Deterministic bid generation problems

[Wang et Xia \(2005\)](#) study a carrier's BGP in the context of TL transportation service procurement. In this paper, the focus is on the bundling method when an OR bidding language is used. They firstly define the bidder's optimality criterion of combinatorial bids. Then two heuristics are developed and compared, one is based on a fleet assignment model and the second is based on the nearest insertion method.

[Lee, Kwon, et Ma \(2007\)](#) consider the carrier's optimal BGP in combinatorial auctions for transportation procurement in TL transportation. Carriers employ vehicle routing models to identify sets of lanes to bid for based on the actual routes. Both column generation and Lagrangian based techniques are used for solving the carrier optimization model and promising results are reported.

Buer (2014) proposes an exact strategy and two heuristic strategies for bidding on subsets of requests. The exact bidding strategy is based on the concept of elementary request combinations. The author shows that it is sufficient for a carrier to bid on each elementary request combination in order to guarantee the same result as bidding on each element of the power-set of the set of tendered requests. The other two heuristic bidding strategies identify promising request combinations, where pairwise synergies based on saving values as well as the capacitated p-median problem are used. The proposed heuristic bidding strategies can help a carrier to increase its chance to win and at the same time can reduce the computational burden to participate in a combinatorial transport auction.

2.4.2.2 Stochastic bid generation problems

O. Ergun, Kuyzu, et Savelsbergh (2007a); Kuyzu et al. (2015) study simultaneous transportation procurement auctions from a truckload carrier's perspective. A stochastic bid price optimization (BPO) problem with the objective of maximizing a carrier's expected profit is formulated in their paper. The formulation takes into account the synergies among the lanes and the competing carriers' bidding strategies. To solve this stochastic optimization problem, they develop an iterative coordinate search algorithm that finds good solutions efficiently. The benefits of employing the BPO technology are demonstrated through computational experiments involving a simulated marketplace.

Triki et al. (2014) deal with a stochastic BGP in the context of long-haul full TL transportation services. They develop a probabilistic optimization model that integrates the bid generation and pricing problems together with the routing planning of the carrier's fleet. Two heuristic procedures

are developed to solve the model with up to 400 auctioned loads.

2.5 Conclusion

In this chapter, we firstly present the background of Collaborative Logistics (CL) or Collaborative Transportation Management(CTM). Then we describe two types of horizontal collaborative logistics : shipper collaboration and carrier collaboration, two types of transportation service : truckload (TL) transportation and less-than-truckload (LTL) transportation. In this thesis, we focus on carrier collaboration in LTL transportation, where two core sub-problems : collaborative transportation planning (CTP) problem and profit allocation problem, should be addressed. We mainly handle CTP in this thesis. At last, a general literature review on CL and centralized planning approaches of CTP, and a detailed literature review on decentralized planning approaches of CTP are given to close the chapter.

Pickup and Delivery Problem with Time Windows, Profits, and Reserved Requests

Contents

| | | |
|------------|---|-----------|
| 3.1 | Introduction | 24 |
| 3.2 | Literature review | 26 |
| 3.3 | Problem description and mathematical model | 28 |
| 3.4 | Adaptive large neighborhood search | 31 |
| 3.4.1 | Initial solution construction | 34 |
| 3.4.2 | Adaptive selection of destroy/repair operators | 36 |
| 3.4.3 | Dynamic adjustment of operator behavior | 38 |
| 3.4.4 | Description of destroy/repair operators | 40 |
| 3.4.5 | Diversification via simulated annealing | 45 |
| 3.4.6 | Local search procedure | 45 |
| 3.5 | Computational experiments | 48 |
| 3.5.1 | Generation of instances | 49 |
| 3.5.2 | Parameter setting | 50 |
| 3.5.3 | Experimental results | 51 |
| 3.5.4 | Impact of the DAOB mechanism | 55 |

3.1 Introduction

This chapter addresses the pickup and delivery problem with time windows, profits, and reserved requests (PDPTWPR), a new vehicle routing problem appeared in carrier collaboration realized through combinatorial auction (CA) or combinatorial exchange (CE). The PDPTWPR is a key sub-problem in the auction-based framework of carrier collaboration.

In carrier collaboration, several carriers form an alliance and exchange some of their transportation requests. Each carrier has a set of *reserved requests* (*i.e.* not proposed for exchange in CA/CE) and can serve additional requests (*selective requests*) acquired from other carriers. Each request is a pickup and delivery request associated with an origin, a destination, a quantity, two time windows (pickup time window and delivery time window), and a price (revenue) for serving the request paid by its corresponding shipper (customer). For each carrier in CA/CE, it has to determine which *selective requests* to bid for, in addition to its *reserved requests*, and builds feasible routes to maximize its total profit. Such a problem raises a new variant of pickup and delivery problem with time windows (PDPTW), *i.e.* the PDPTWPR. To the best of our knowledge, this problem was rarely studied in the literature.

A mixed-integer programming (MIP) model is formulated for the PDPTWPR and an improved version of adaptive large neighborhood search (ALNS) approach well adapted to the problem is developed. The ALNS involves ad-hoc destroy/repair operators and a post-optimization

procedure conducted by local search (LS). The algorithm runs in successive segments which change the behavior of operators and compute their own statistics to adapt selection probabilities of operators. The MIP model and the ALNS approach are evaluated on 54 randomly generated instances with up to 100 requests. The computational results indicate that the ALNS significantly outperforms the CPLEX MIP solver, not only in terms of solution quality but also in terms of CPU time.

Therefore, the contributions of this chapter are summarized as follows :

1. The PDPTWPR, a new variant of vehicle routing problem, is proposed under the context of LTL carrier collaboration.
2. An improved version of adaptive large neighborhood search (ALNS) approach well adopted to the PDPTWPR is developed.
3. A set of destroy/repair operators are designed for the ALNS algorithm according to the property of the PDPTWPR.
4. A mechanism which can dynamically adjust the operators' behavior to be conservative/aggressive is firstly used for routing problems with profits.

This chapter is organized as follows. A brief literature review of vehicle routing problems (VRP) and vehicle routing problems with profits (VRPP) is provided in Section 3.2. Section 3.3 describes the PDPTWPR and provides its MIP model. An improved version of ALNS to solve the PDPTWPR is developed in Section 3.4. Section 3.5 presents numerical experiments which compares the results of our ALNS algorithm with the ones obtained by the CPLEX MIP solver on the randomly generated instances. Finally, Section 3.6 closes this chapter with some remarks for future research.

3.2 Literature review

Our problem is related to the pickup and delivery problem with time windows (PDPTW), which itself is a generalization of the vehicle routing problem with time windows (VRPTW). The PDPTW involves three main constraints : time window constraints, capacity constraints and coupling constraints (the delivery node of each request must be visited after its corresponding pickup node in the same route). The PDPTW has been well studied in the literature and due to its complexity, metaheuristic algorithms have become dominating methods for its resolution. [Nanry et Wesley Barnes \(2000\)](#) propose a reactive tabu search and test it on instances with up to 50 requests. [H. Li et Lim \(2003\)](#) create a set of benchmark instances and propose a hybrid metaheuristic. [Hosny et Mumford \(2012\)](#) compare sequential and parallel insertion heuristics to provide metaheuristics with high quality initial solutions. [Bent et Hentenryck \(2006\)](#) apply Variable Neighborhood Search (VNS) to the PDPTW and their computational results show promising performance of their algorithm, compared with the previous PDPTW metaheuristics. [Ropke et Pisinger \(2006\)](#) design an ALNS algorithm which is probably the most effective metaheuristic for the PDPTW so far, with results reported for up to 1000 locations.

Our PDPTWPR displays important differences with the PDPTW : i) serving all requests is not mandatory (provided all reserved requests are treated), ii) a profit is associated with each request, and iii) the objective function, to be maximized, is the sum of the revenues minus the routing costs. We find no reference on this problem in the literature, although a growing number of publications deals with vehicle routing problems with profits (VRPP) in general.

Single-vehicle problems with profits are surveyed in Feillet, Dejax, et Gendreau (2005). Tour costs and collected profits can be expressed in the objective function, by minimizing the travel costs minus the profits, giving the profitable tour problem (PTP). The profits collected can be maximized, subject to a maximum tour length, which defines the orienteering problem (OP). Conversely, in the prize-collecting traveling salesman problem (PCTSP) (Balas, 1989), the travel costs are minimized but the collected profits cannot be less than a given constant.

Among these problems, the PTP has the same objective function as our PDPTWPR. Only heuristics are available to solve it. Nguyen et Nguyen (2010) develop an approximation algorithm, based on the heuristic from Frieze, Galbiati, et Maffioli (1982) for the asymmetric traveling salesman problem (ATSP), and a method to round fractional solutions of a linear programming relaxation for the asymmetric PTP. Goemans et Bertsimas (1990) solve an undirected version of the PTP.

Routing problems with multiple vehicles and profits are much less studied. Butt et Cavalier (1994) define the multiple tour maximum collection problem (MTMCP), a generalization of the OP where the same maximum tour length is applied to several vehicles. Chao, Golden, et Wasil (1996) study the same problem but introduce a nowadays standard name, the team orienteering problem (TOP). A few recent papers have tackled the TOP with time windows (TOPTW), see for instance Labadie, Mansini, Melchovsky, et Wolfler-Calvo (2012) who develop a granular variable neighborhood search. The TOPTW is close to our PDPTWPR but does not distinguish between pickup and delivery nodes. A recent paper by Archetti, Corberan, Sanchis, Plana, et Speranza (2014) present the team orienteering arc routing problem (TOARP), but in a truckload (TL) context : since each

vehicle can transport one request at a time, each request can be modeled by one arc, which leads to an arc routing problem (ARP).

For more details on VRPPs, we refer readers to the technical report written by [Archetti, Speranza, et Vigo \(2013\)](#).

3.3 Problem description and mathematical model

The PDPTWPR is based on a complete undirected graph $G = (N, E)$. The node-set is defined as $N = \{0, \dots, 2n+1\}$, where n denotes the number of requests. Nodes 0 and $2n + 1$ represent the depot of the carrier, hosting a set $K = \{1, \dots, m\}$ of m identical vehicles of capacity Q . It is assumed that each vehicle route begins at node 0 and ends at node $2n + 1$. Each node i has a time window $[a_i, b_i]$ to begin service, while each edge (i, j) in E has a travel cost c_{ij} and a travel time t_{ij} . The service times at node i is included in the t_{ij} 's. Like in the VRPTW, a vehicle can wait at customer i if it arrives before a_i . The subset $P = \{1, \dots, n\}$ contains the pickup nodes of all requests, while $D = \{n + 1, \dots, 2n\}$ gathers delivery nodes. Request $i, i = 1, \dots, n$, is associated with a pickup node i , a delivery node $n + i$, a demand $d_i > 0$ and a price p_i (customer payment). For the delivery node, we set $d_{n+i} = -d_i$. The set R of all requests includes the subset of reserved requests R_r and the subset of selective requests R_s .

The goal is to determine the selective requests to be served, in addition to the reserved requests, and to determine the associated vehicle routes, to maximize the total profit which is equal to the sum of collected payments minus the total cost of the routes. The demand served in a route cannot exceed vehicle capacity, the time window at each node must be respected, and the delivery node of each request must be visited after its correspon-

ding pickup node, in the same route.

The problem is NP-hard in strong sense like the PDPTW which is the particular case where R_s is empty and all prices p_i are equal to a large positive constant M (to ensure that all requests are served).

The PDPTWPR can be formulated by a MIP model. In addition to previous data, we need two symbols to write the model more easily : $T_{ij} = b_j - a_i$ plays the role of a big-M constant in the time window constraints, while $Q_i = Q + d_i$ is used in the capacity constraints. The following decision variables are also used in the model :

- x_{ij}^k , binary variable equals to 1 if and only if vehicle k travels directly through arc (i, j) ,
- y_i^k , binary variable equals to 1 if and only if request i is served by vehicle k ,
- T_i^k , time at which vehicle k begins service at node i ,
- Q_i^k , load of vehicle k when leaving node i .

Resulting model :

$$\max \sum_{k \in K} \sum_{i \in R} p_i \cdot y_i^k - \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} \cdot x_{ij}^k \quad (3.1)$$

Subject to :

$$\sum_{j \in N, j \neq i} x_{ji}^k - \sum_{j \in N, j \neq i} x_{ij}^k = 0 \quad \forall i \in P \cup D, \forall k \in K \quad (3.2)$$

$$\sum_{j \in P, j \neq 0} x_{0j}^k = 1 \quad \forall k \in K \quad (3.3)$$

$$\sum_{i \in D, i \neq 2n+1} x_{i, 2n+1}^k = 1 \quad \forall k \in K \quad (3.4)$$

$$\sum_{k \in K} y_i^k = 1 \quad \forall i \in R_r \quad (3.5)$$

$$\sum_{k \in K} y_i^k \leq 1 \quad \forall i \in R_s \quad (3.6)$$

$$\sum_{j \in N, j \neq i, 2n+1} x_{ij}^k = y_i^k \quad \forall i \in P, \forall k \in K \quad (3.7)$$

$$\sum_{j \in N, j \neq i, 0} x_{j,n+i}^k = y_i^k \quad \forall i \in P, \forall k \in K \quad (3.8)$$

$$T_i^k + t_{i,n+i} \leq T_{n+i}^k \quad \forall i \in P, \forall k \in K \quad (3.9)$$

$$T_j^k \geq T_i^k + t_{ij} \cdot x_{ij}^k - T_{ij} \cdot (1 - x_{ij}^k) \quad \forall i, j \in N, \forall k \in K \quad (3.10)$$

$$a_i \leq T_i^k \leq b_i \quad \forall i \in N, \forall k \in K \quad (3.11)$$

$$Q_j^k \geq Q_i^k + d_j - Q_j \cdot (1 - x_{ij}^k) \quad \forall i, j \in N, \forall k \in K \quad (3.12)$$

$$\max\{0, d_i\} \leq Q_i^k \leq \min\{Q, Q + d_i\} \quad \forall i \in N, \forall k \in K \quad (3.13)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in N, \forall k \in K \quad (3.14)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in R, \forall k \in K \quad (3.15)$$

$$T_i^k \geq 0 \quad \forall i \in N, \forall k \in K \quad (3.16)$$

$$Q_i^k \geq 0 \quad \forall i \in N, \forall k \in K \quad (3.17)$$

The objective function (3.1) represents the total profit of the carrier, equals to the difference between the sum of payments of served requests and the total transportation cost. Constraints (3.2) ensure that a vehicle arriving at a pickup or delivery node has to leave it. Constraints (3.3) and (3.4) indicate that each vehicle leaves the depot and returns to it. Constraints (3.5) guarantee that all reserved requests must be served once, whereas in constraints (3.6) selective requests are served at most once. Constraints (3.7) and (3.8) ensure that if a request is served, there must be a vehicle leaving its pickup node and arriving at its pairwise delivery node. Time win-

dows and precedence relations are respected via constraints (3.9) to (3.11). Constraints (3.12) and (3.13) concern vehicle capacity. Finally, constraints (3.14) to (3.17) define the variables.

3.4 Adaptive large neighborhood search

For classical vehicle routing problems (without profits), the small-scale moves used in local search procedures affect the partition of customers in routes and the sequence of these routes. In problems with profits, other moves are required to modify the set of served requests since it is not mandatory to serve all of them. As the possibility of choosing requests tremendously expands solution space, the two kinds of moves must be combined in a clever way to avoid excessive running time. For instance, [Labadie et al. \(2012\)](#) propose a VNS algorithm for the TOPTW where a local search procedure focuses on route sequences, while the shaking step changes the subset of served requests. For the PDPTWPR, we select the adaptive large neighborhood search (ALNS) framework as another way to remedy the weak efficiency of small-scale neighborhoods.

The precursor of the ALNS is Large Neighborhood Search (LNS), introduced by [Shaw \(1998\)](#) for the capacitated vehicle routing problem (CVRP). LNS begins with an initial solution and improves the objective value gradually, by applying one destroy and one repair operator at each iteration. The destroy operator is a randomized heuristic removing a small subset of customers. The repair operator reinserts these customers optimally, using constraint programming and branch-and-bound, see [Bent et Hentenryck \(2006\)](#) for the VRPTW. The destroy and repair operators are also called ruin and recreate operators, or removal and insertion operators.

The application of a destroy/repair pair can be viewed as a move that implicitly defines a very large neighborhood. However, only one move is randomly selected at each iteration instead of exploring the neighborhood completely. LNS is conceptually simple but has some known drawbacks. The search is a bit blind because the destroy/repair moves sample a very small fraction of the large neighborhood. This can be compensated by more iterations but, added to the exact method used to reinsert customers, the metaheuristic becomes time-consuming.

Ropke et Pisinger (2006) propose an ALNS to improve the LNS. The ALNS involves several destroy and repair operators, which are all heuristics to achieve a time-saving purpose. At each iteration, a pair of operators is randomly chosen to make a move and statistics are computed to favor the most efficient pairs. The method is adaptive since the most frequent pairs may change during the search. The ALNS has been successfully applied to the PDPTW (Ropke & Pisinger, 2006) and later to various rich vehicle routing problems (Pisinger & Ropke, 2007; Aksen, Kaya, Salman, & Tüncel, 2014).

Our ALNS is motivated by Ropke et Pisinger (2006), but we bring six important modifications to cope with the peculiarities of the PDPTWPR so as to achieve a good efficiency. The first one is the design of specific destroy/repair operators, which acts both on the sequence of the routes and on the selection of served requests. The second change is to restart the ALNS from several initial solutions. The iterations for one initial solution define a *run*. The third modification is to organize the search in what we call successive *segments*. The behavior of operators is modified at each new segment. At the beginning of the ALNS, both *reserved requests* and *selective requests* can be removed and inserted, even if it is not profitable. Then, gradually,

less and less *reserved requests* can be removed out and more and more profitable insertions are preferred. The fourth change is a diversification technique called meta-destroy : When the search is stalled, moves combining two destroy and one repair operators are tried. The two last changes are not really new since they are used in some recent ALNS implementations in [Aksen et al. \(2014\)](#) : a local search procedure is added for intensification and the simple descent is replaced by a simulated annealing (SA) loop.

The general structure of our algorithm is sketched in Algorithm 1 and its components are detailed in the following subsections. The main loop performs $nruns$ independent runs to find a global best solution S^* and its costs $f(S^*)$. Each run calls a sequential insertion heuristic : *SIH* (Subsection 3.4.1) which initializes the current best solution S_{best} of the run, sets the SA temperature T to its initial value T_{beg} and initializes the weights and scores of operators. The weight and score system is explained in Subsection 3.4.2 while the different destroy/repair operators used are presented in Subsection 3.4.4. The second loop executes $nsegs$ successive segments. Each segment selects different possible behaviors of destroy/repair operators, as explained in subsection 3.4.3, performs $niters$ the ALNS iterations (third loop) and ends by calling a local search procedure *LS* (Subsection 3.4.6) and updating the global best solution in case of improvement. Each ALNS iteration selects a combination of operators and executes the corresponding move and improves the current solution or satisfies the SA criterion, where $r^{U(0,1)}$ is random value between 0 and 1 generated according to the uniform distribution $U(0, 1)$. The SA scheme is commented in Subsection 3.4.5.

Algorithm 1 – Pseudo code of our ALNS metaheuristic : $ALNS(S^*)$

```

1:  $f(S^*) \leftarrow \infty$ 
2: for  $run \leftarrow 1$  to  $nruns$  do
3:   call  $SIH(S)$  (Algorithm 2)
4:    $S_{best} \leftarrow S$ 
5:    $T \leftarrow T_{beg}$ 
6:   initialize weights and scores of operators (Subsection 3.4.2)
7:   for  $seg \leftarrow 1$  to  $nsegs$  do
8:     select the behavior of operators (Subsection 3.4.3)
9:     for  $iter \leftarrow neters$  do
10:      select one destroy operator and one repair operator or two destroy operators if  $S_{best}$  is not improved in the last consecutive  $\delta$  iterations of the run (Subsection 3.4.4)
11:      apply selected move  $S \rightarrow S'$ 
12:      if  $S'$  feasible and  $(f(S') > f(S))$  or  $r^{U(0,1)} < \exp((f(S')-f(S))/T)$  (Subsection 3.4.5) then
13:         $S \leftarrow S'$ 
14:        update performance scores of selected operators (Subsection 3.4.2)
15:        if  $f(S) > f(S_{best})$  then  $S_{best} \leftarrow S$  endif
16:      end if
17:       $T \leftarrow T \times \theta$ 
18:    end for
19:    update the weights of the ALNS moves and reset scores (Subsection 3.4.2)
20:    call  $LS(S)$  (Subsection 3.4.6)
21:  end for
22:  if  $f(S_{best}) > f(S^*)$  then  $S^* \leftarrow S_{best}$  endif
23: end for

```

3.4.1 Initial solution construction

According to our experimental results, the quality of the initial solution may have a crucial impact on the final outcome of the ALNS. Consequently, we develop an effective sequential insertion heuristic (SIH) to provide quickly each run with a high-quality initial solution. SIH can be controlled by using three insertion policies :

- *Policy 1* : Only the *reserved requests* are served.
- *Policy 2* : After the insertion of all *reserved requests*, only profitable *selective requests* can be inserted. The insertion procedure will stop if no profitable *selective request* can be found any more.
- *Policy 3* : After the insertion of all *reserved requests*, *selective requests* are inserted until infeasibility.

Policy 1 is the most basic way to construct a feasible initial solution, by including the whole set of *reserved requests*. In general, *Policy 2* produces a better initial solution than the others two, but we observe that in a few cases that a near-optimal initial solution may lead to be trapped in a local optimum at the very beginning of the ALNS searching process. In comparison with the two first policies, *Policy 3* tends to exhaust vehicle fleet capacity. For each run of the ALNS, *SIH* randomly select *Policy 1, 2* or *3* with respective probabilities γ_1 , γ_2 and γ_3 , with $\gamma_1 + \gamma_2 + \gamma_3 = 1$.

As shown in Algorithm 2, *SIH* builds one route at a time. The *reserved requests* are first sorted in decreasing order of price and marked as unserved. The sorted list is browsed and the existence of at least one feasible insertion slot is checked for the current request i . If feasible insertion places exist in the current route, the request is marked as served and its two nodes i and $n + i$ are inserted into the most profitable position. If no feasible insertion is possible, the request remains unserved in the list and will be tested again in a new route. When no request can be inserted, a new route is initiated. This process is repeated until all *reserved requests* are served. The *selective requests* are treated in the same way by taking *Policy 2* or *3* into account.

Algorithm 2 – Sequential insertion heuristic – $SIH(S)$

```
1: sort reserved requests in decreasing order of prices in a list  $L$ 
2: set current route index  $r$  to 0
3: mark all reserved requests as unserved
4: repeat
5:    $r \leftarrow r + 1$ 
6:   initialize a new route using the 1st unserved request of  $L$ 
7:   for all unserved request  $i$  in  $L$  do
8:     if feasible insertions exist in route  $r$  for  $i$  then
9:       insert  $i$  in the most profitable position
10:      mark  $i$  as served
11:     end if
12:   end for
13: until all reserved requests are served or  $r = m$  (maximum fleet capacity
    reached)
14: if Policy  $\neq 1$  then
15:   repeat steps 1-13 but for the selective requests and follow the Policy 2
    or 3
16: end if
```

3.4.2 Adaptive selection of destroy/repair operators

At each iteration, the ALNS algorithm employs one or two removal operators to partially destroy the current solution and then repairs it by utilizing one insertion operator. One question is how to select these operators more effectively. Like other ALNS implementations in the literature, the algorithm chooses the most suitable combination of operators depending on their past performance. For diversification purposes, poor-performance operators still need to have a low selection probability to be selected during the search. We use a roulette-wheel mechanism. Assuming that n operators are available, each operator is associated with a weight ω_i which reflects its performance during its previous outcomes. Each operator j is randomly selected with probability $\omega_j / \sum_{i=1}^n \omega_i$ for the current iteration.

The weight ω_i of each operator i is set to 1 at the beginning of each

ALNS run. It remains fixed during the iterations of a segment but it is adjusted at the end of the segment on the basis of a performance score. At the beginning of each segment, all scores are initialized to 0, for the reason that low performance operators can still have a chance to be selected even if they were seldom selected in the previous segment. For feasible moves, the scores are updated by adding either ϱ_1 , ϱ_2 , ϱ_3 or ϱ_4 according to the four different situations in Table 3.1.

In practice, the score adjustment parameters should be set such that $\varrho_1 > \varrho_2 > \varrho_3 > \varrho_4$. The first situation is highly rewarded since it yields a new best solution. The second and third situations are still interesting because the current solution is improved. We prefer to favor the second condition because finding a solution with new characteristics means that the search is driven to an unexplored area of solution space. To detect a new solution, the past solutions are stored. To achieve efficient comparison, only the following characteristics of a past solution are stored : total profit, the number of vehicle used and the number of customers served by each vehicle. The pool is kept sorted in increasing profit order and the existence of a solution with a given cost is checked using dichotomic search. The awarding of score ϱ_4 based on the SA acceptance criterion is used to prevent the search from looping on the same operators and also to bring some diversification.

TABLE 3.1: Adaptive adjustment of the operator scores

| Increment | Conditions on the solution obtained by the operators |
|-------------|--|
| ϱ_1 | A new best solution is obtained. |
| ϱ_2 | The solution is not found before and improves the current solution. |
| ϱ_3 | The solution is not a new one but improves the current solution. |
| ϱ_4 | The solution is worse than the current solution but still accepted by the SA scheme. |

To update the operator weights after each segment seg , let $\omega_{i,seg}$ be the weight of operator i used in the segment, $\alpha_{i,seg}$ the number of times the

operator was called, $\beta_{i,seg}$ its resulting score, and $\eta \in [0, 1]$ a reaction factor representing how quick the weights react to performance. The weights are adjusted using formula (3.18). Note that if $\eta = 1$ then the previous weight is completely ignored and the new weight solely depends on the score achieved in the last segment. The other extreme is $\eta = 0$ which preserves the current weight while ignoring the score.

$$\omega_{i,seg+1} = \begin{cases} \omega_{i,seg} & \text{if } \beta_{i,seg} = 0 \\ (1 - \eta) \cdot \omega_{i,seg} + \eta \cdot \beta_{i,seg} / \alpha_{i,seg} & \text{otherwise} \end{cases} \quad (3.18)$$

3.4.3 Dynamic adjustment of operator behavior

The PDPTWPR is highly complex since it combines the choice of selective requests and routing decisions. Even a minor modification on the current solution might deeply affect final results. To improve final solution quality and to vary the number of served requests, we implemented a technique called *Dynamic Adjustment of Operator Behavior* (DAOB). The basic idea is to modify progressively the behavior of operators over the successive segments of the ALNS. Firstly, we develop one group of removal policies and two groups of insertion policies based on the specific features of the PDPTWPR.

Request Remove Policies (RRP) :

1. Both *selective requests* and *reserved requests* are removable.
2. Only *selective requests* are removable.

Request Insertion Priority Policies (RIPP) :

1. *Selective requests* and *reserved requests* have the same priority.

2. All *reserved requests* must be served before treating *selective requests*.

Insertion Threshold of Selective Requests (ITSR) :

1. *Selective requests* are inserted regardless of profitability.
2. *Selective requests* are inserted only if they are profitable (insertion cost < service payment).

We would refer to the policies marked by '1.' as Code-1 policies, and the policies marked by '2.' as Code-2 policies.

As explained in the next subsection describing each operator, each ALNS iteration applies one or two destroy operators to remove a given number of requests, and then one repair operator. The destroy operators can remove any request in *Policy RRP1* but only *selective requests* in *RRP2*. While the ALNS for the PDPTW (Ropke & Pisinger, 2006) reinserts all removed requests (since all requests must be served), our repair operators try to insert unserved requests according to the selected insertion policies (*RIPP1/RIPP2* and *ITSR1/ITSR2*), as long as feasible insertions exist. An infeasible solution, *i.e.*, a solution violating certain constraints or with negative profit, can be obtained if some *reserved requests* cannot be served, due to time windows or lack of vehicle capacity.

Clearly, Code-1 policies allow deep changes in the current solution and induce long-range moves while the more conservative policies with Code-2 favour the generation of feasible solutions and a faster improvement of the objective function. At the beginning of our ALNS (first segment) it is worthwhile to widen the search by using more frequently Code-1 policies. However, because of *Policy RIPP1*, infeasible solutions are frequent since *selective requests* may exhaust vehicle capacity and leave a few *reserved requests* unserved. Also, *Policy ITSR1* results in very slow improvements on

the objective function.

Hence, the probability of using Code-2 policies is gradually augmented at the beginning of each segment. In the last segments, the ALNS generates more feasible solutions and tends to faster improve the total profit. Let P_1 and P_2 denote the probabilities of using policies with Code-1 or Code-2, respectively. They are simply computed as $P_1 = 1 - P_2$ and $P_2 = seg/nsegs$ (where seg is the current segment number and $nsegs$ the total number of segments, $nsegs > 2$).

To better understand the DAOB, the change of neighborhoods N_1, N_2, \dots for three cases is depicted in Fig. 3.1. The first one illustrates a traditional ALNS applied to the PDPTW : most of the solution space is searched. The second one corresponds to the first segment of our DAOB for the PDPTWPR. The destroy/repair operators define wider neighborhoods which include many infeasible solutions (shown in grey). The last situation corresponds to the more conservative operators in the last segments : neighborhood are reduced and focus on feasible solutions.

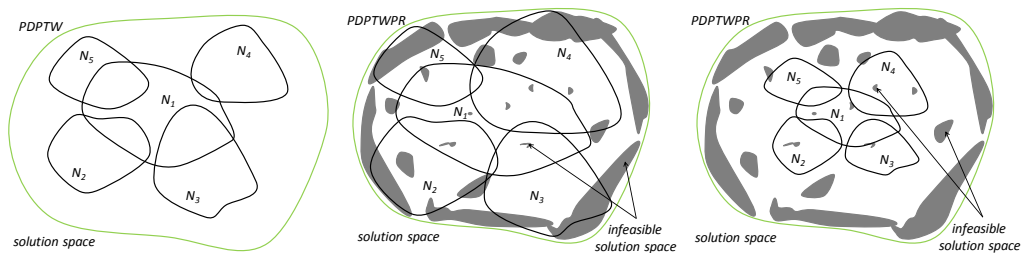


FIGURE 3.1: Illustration of neighborhoods in a classical ALNS and in our DAOB version.

3.4.4 Description of destroy/repair operators

Although some operators in our ALNS are similar to the ones designed in [Ropke et Pisinger \(2006\)](#) for the PDPTW, they must be adapted to serve a

variable number of *selective requests* and deal with profits. For non-selective VRPs, all requests must be served and one destroy operator removes a certain number of requests which must be reinserted by another repair operator. Conversely, in the PDPTWPR, partial solutions are already feasible as long as all *reserved requests* are served. Moreover, the behavior of our destroy and repair operators (different treatment for *reserved requests* and *selective requests*) is affected by the DAOB policies explained in the previous subsection.

We also implement a meta-destroy operator to diversify the search when a maximum number δ of successive iterations without improving best solution is reached. This mechanism is independent from the decomposition of the search into segments : The number of iterations without improvement is counted from the beginning of each ALNS run and the maximum number is checked at each iteration of each segment. The meta-destroy operator consists in applying two destroy operators instead of one.

The number of routes, or vehicles actually used, \bar{m} can be modified by our operators. When one destroy operator removes the only request from a route, this route is closed. When looking for a best insertion for a request, the repair operators consider the \bar{m} non-empty routes plus, if $\bar{m} < m$, one "empty" route reduced to the two depot nodes 0 and $2n + 1$.

3.4.4.1 Destroy operators

Our ALNS involves six destroy operators described. Given the number \bar{n} of the requests in the incumbent solution and a removal fraction $\rho \in [0, 1]$, each of the operators applies a strategy to select $\lceil \rho \cdot \bar{n} \rceil$ requests (among all requests in *Policy RPP1*, or only among *selective requests* in *RPP2*). These re-

quests are then removed from the routes. Only one destroy operator is executed in each ALNS iteration, except in the meta-destroy scheme (applied after δ successive non-improving iterations) where two destroy operators are applied to bring diversification.

Random removal. This operator randomly selects the $\lceil \rho \cdot \bar{n} \rceil$ requests to be removed. Depending on the value of ρ , it may significantly modify the incumbent solution.

Least profit removal. The profit of a request i served in the incumbent solution S is defined as $f(S) - f(S')$, where $f(S')$ is the objective function without request i . The least profit removal sketched in Algorithm 3 removes $\lceil \rho \cdot \bar{n} \rceil$ requests with low profits, because they might often be reinserted in more profitable positions. The operator is randomized to avoid repeatedly removing the $\lceil \rho \cdot \bar{n} \rceil$ requests with lowest profits.

Algorithm 3 – Least profit removal

- 1: $removed \leftarrow 0$
 - 2: **repeat**
 - 3: sort all requests (*Policy RPP1*) or all *selective requests* (*Policy RPP2*) served in S in increasing order of profit in an array L
 - 4: compute a random request index $j = \lceil (r^{U(0,1)})^{100 \cdot \rho} |L| \rceil$
 - 5: remove request L_j from S
 - 6: $removed \leftarrow removed + 1$
 - 7: **until** $removed = \lceil \rho \cdot \bar{n} \rceil$
-

Least paid removal. The least paid removal operator has the same algorithm as the least profit removal, except that array L is now sorted in increasing order of prices p_i . Considering request prices is essential in the PDPTWPR : removing the cheapest requests and putting them in the request pool (case of *selective requests*) or reinserting them in other positions

may lead to a better solution.

Most expensive removal. Given a request i served in the incumbent solution S , we define its cost as $f(S) - f(S')$, where $f(S) - f(S')$ represents the difference of transportation cost with or without request i . This operator is widespread in ALNS metaheuristics for general VRPs. It works like the least profit and least paid removal (Algorithm 3), except that array L is sorted in decreasing order of requests' cost.

Shaw removal. We use the same way of implementing Shaw removal Shaw (1998) as Ropke et Pisinger (2006). Firstly, a seed request is chosen randomly and the heuristic removes similar requests in terms of distance (a request whose pickup and delivery nodes are close to those of the seed request is favored), time (starts of service at the two nodes are similar in the two requests), and demands. It is also applied $\lceil \rho \cdot \bar{n} \rceil$ times with the same randomization as in Algorithm 3. The underlying idea is that similar requests less frequently violate capacity and time window constraints when they are reshuffled around in groups.

Price similarity removal. This operator is similar to the previous one but it removes requests which are similar in terms of price. Then a repair operator will exchange their locations or directly abandon them to increase total profit. We use in fact a dissimilarity measure for two requests i and j , defined as their price difference $P(i, j) = |p_i - p_j|$: growing values correspond to more and more dissimilar prices. This operator is outlined in Algorithm 4.

Algorithm 4 – Price similarity removal

-
- 1: randomly select one seed-request r from solution S and put it in a set Z
 - 2: **while** $|Z| < \lceil \rho \cdot \bar{n} \rceil$ **do**
 - 3: sort requests of $S \setminus Z$ such as $i < j \Rightarrow P(r, i) < P(r, j)$ in a list L
 - 4: compute a random request index $j = \lceil (r^{U(0,1)})^{100 \cdot \rho} |L| \rceil$
 - 5: $Z = Z \cup \{L_j\}$
 - 6: **end while**
 - 7: remove all requests of Z from S
-

3.4.4.2 Repair operators

Two repair operators were utilized in our ALNS. Their behavior depends on the *policies* *RIPP1/2* and *ITSR1/2* selected by the DAOB. They insert unserved requests (not only those removed by the destroy operators) as long as feasible insertions are possible.

Basic greedy insertion heuristic. This greedy heuristic inserts one by one unserved requests. The two nodes i and $n + i$ of request i are inserted in order to achieve the largest increase of total profit.

Regret insertion heuristic. The basic greedy heuristic seems quite myopic as it only considers the profit change of one request : the later an attempt of a request insertion is made, the more difficult it is to insert this request at a good insertion position (slot), because the insertion of other requests reduced the number of possible insertion slots. The regret insertion heuristic tries to anticipate by computing for each unserved request a regret value equal to the total profit difference between the best insertion and the second best one. Thus, one request with a high regret value will be inserted firstly. The regret values must be recomputed after each insertion, because

some insertion positions are no longer available.

3.4.5 Diversification via simulated annealing

The simulated annealing (SA) scheme appears clearly in the main algorithm. Its goal is to avoid to be trapped in a local optimum. Compared with a descent heuristic which only accepts improved solutions, the SA accepts a degrading move $S \rightarrow S'$ (when $f(S') < f(S)$) with a probability $e^{-(f(S)-f(S'))/T}$. The probability decreases with the profit disparity and with parameter T called temperature.

At the beginning of each run, T is set to T_{beg} . A number computed to accept a solution 30% worse than the initial solution with a given probability τ . The temperature is reduced after each iteration (over successive segments) by multiplying T a cooling factor $\theta \in (0, 1)$. In practice, θ must be close to 1 to achieve a slow cooling.

3.4.6 Local search procedure

Observe that embedding a local search procedure in our ALNS is often beneficial to improve the outcome of metaheuristic. Consequently, we decide to implement six moves in a local search procedure called only at the end of each segment to keep running time at a reasonable level.

The moves are inspired by the ones of the classical VRP literature (Toth & Vigo, 2014) with the difference that we relocate or exchange pairs of nodes. The moves are illustrated in Fig. 3.2 to Fig. 3.7. Circles represent pickup nodes while triangles denote delivery nodes. *Reserved requests* are filled in black while *selective requests* have an empty interior. The *2-Opt* move is not included because time windows make it infeasible in most

cases.

The local search procedure works as follows. Three types of moves are randomly selected, including at least one of the two selective moves (selective request removal and selective request insertion) for the reason that they are more effective than the other four in most cases. The neighborhoods defined by the three types are searched in the order of description below. All feasible moves in the incumbent neighborhood are tested. If improving moves are found, the best one is executed and the neighborhood is examined again, otherwise the search proceeds with the next neighborhood type. The local search stops when the last type yields no improvement.

- *Intraroute relocate* : One pickup node or a delivery node is removed to be reinserted in another position of the same route (Fig. 3.2).
- *Interoute relocate* : One request is removed from one route and reinserted in another (Fig. 3.3).
- *Intraroute exchange* : Two requests are exchanged in the same route (Fig. 3.4).
- *Interoute exchange* : Two requests are exchanged between two routes (Fig. 3.5).
- *Selective request removal* : One *selective request* is removed from its route and becomes unserved (Fig. 3.6).
- *Selective request insertion* : One unserved *selective request* is inserted in a route (Fig. 3.7).

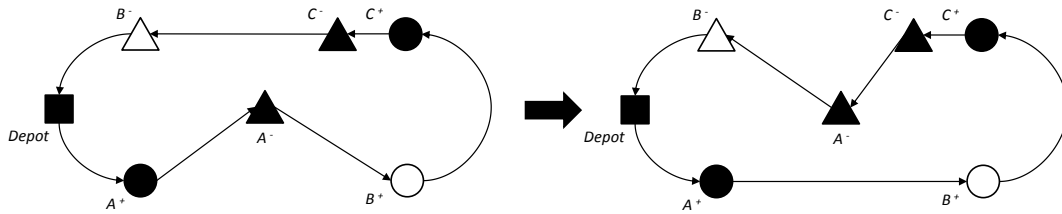


FIGURE 3.2: Intraroute relocate : delivery node of request A is relocated

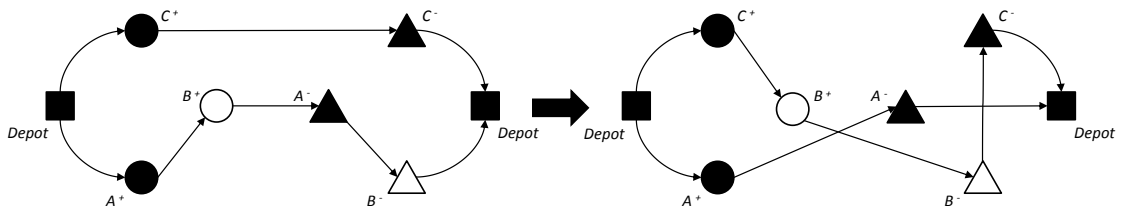


FIGURE 3.3: Interroute relocate : request B is relocated

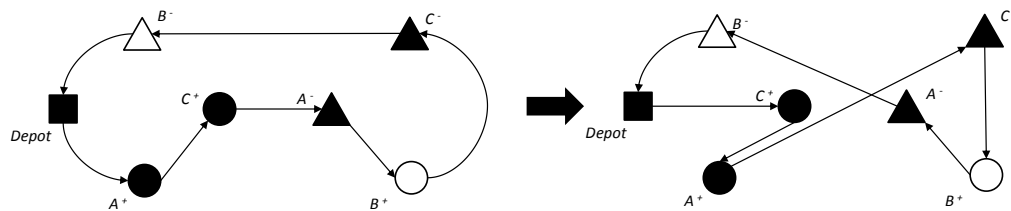


FIGURE 3.4: Intraroute exchange : request A and request C are exchanged

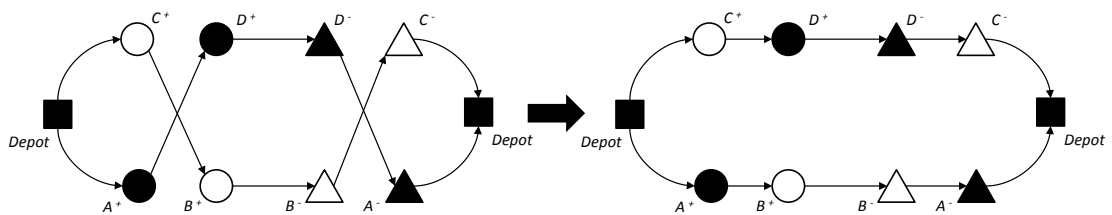


FIGURE 3.5: Interroute exchange : request B and request D are exchanged

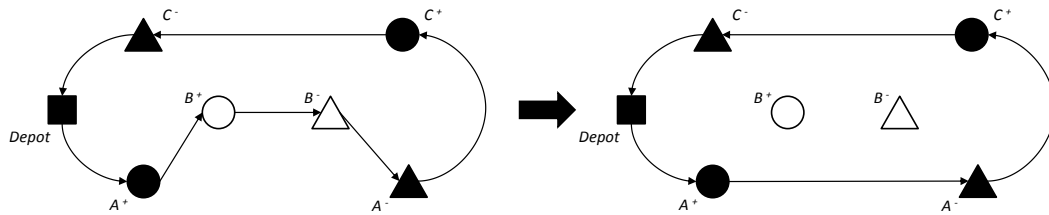


FIGURE 3.6: Selective requests removal : request B is removed

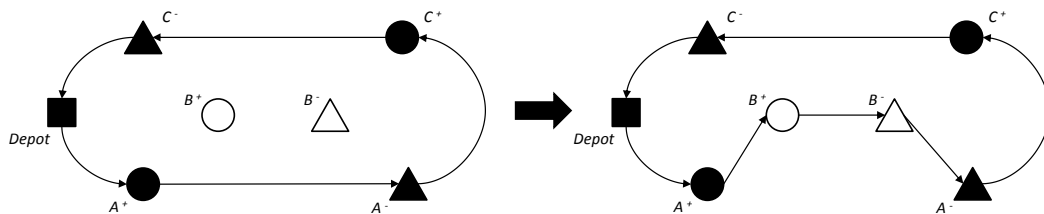


FIGURE 3.7: Selective requests insertion : request B is inserted

3.5 Computational experiments

To evaluate the performance of the ALNS algorithm, we generate 54 instances partitioned in small size ($n \in \{10, 20\}$), medium size ($n \in \{30, 40, 50\}$) and large size ($n = 100$) for the PDPTWPR. The ALNS is compared with the CPLEX MIP solver (version 12.6).

The following subsections describe the generation of instances, list the parameter values used in our algorithm, and provide test results and optimality gaps which are reported separately for small, medium and large size instances. An analysis on the percentage of infeasible solutions and the impact of the DAOB mechanism closes the section.

3.5.1 Generation of instances

The instances of this study are generated based on the Euclidean benchmark instances proposed by [Ropke et Pisinger \(2006\)](#) for the PDPTW, available at URL <http://www.diku.dk/~sropke/>. We copy the instances with the coordinates of each node, the demand and time windows of each request.

Each instance name has a format $n - |R_r| - |R_s| - source$. Consider instance 10-5-5-50a as an example. There are 10 requests in total, including 5 reserved requests (1-5) and 5 selective requests (6-10). The code 50a means this instance is derived from the original one, prob50a : Only the 10 first requests appearing in prob50a are copied in instance 10-5-5-50a.

Some specific data required for the PDPTWPR are added. For each instance size (10, 20, 30, 40, 50 or 100 requests), we build nine instances using nine PDPTW files. These nine instances can be decomposed in three types : three with roughly one-third of reserved requests, three with 50% of reserved requests, and three with two-thirds of reserved requests.

The fleet size of original instances is regarded as a reference. It is adjusted in accordance with the proportion of the number of requests extracted from the original instance.

The service price of each request is set according to the coordinates of its nodes. Take request i as an example, let $d_{i,n+i}$ denote the distance from its pickup node i to delivery node $n + i$, then this request is given a service price $p_i = d_{i,n+i} \cdot \lambda$, $\lambda \in [3, 5]$. This formula generates a large proportion of profitable requests and a small proportion of non-profitable requests.

Finally, since some original instances have multiple depots, we select one of the depots arbitrarily in such cases.

3.5.2 Parameter setting

3.5.2.1 Computational environment and CPLEX setting

All experiments were conducted on a desktop equipped with Intel(R) Core (TM) i7-2600 3.40 gigahertz processor and 8 gigabyte of RAM. The operating system of this PC is 64-bit Window 7. The ALNS algorithm was coded in C++ using the development platform Visual Studio 2013. The Optimization Programming Language (OPL) and the CPLEX solver 12.6 were used to solve the MIP model. CPLEX 12.6 was called with the following option settings : $nodefileind = 2$, $workmem = 128$, $memoryemphasis = 1$, $threads = 8$, $nodesel = 2$ and $varsel = 3$ (see CPLEX 12.6 Solver Manual). With these options, the computation load of CPLEX is distributed over the four cores (8 threads) of the Intel(R) Core (TM) i7-2600 processor.

We solved the MIP model with a preset time of 0.5, 1, 1.5, 2, 4 hours (1800, 3600, 5400, 7200, 14400 seconds) with 10, 20, 30, 40, 50 requests, respectively. For the large size instances with 100 requests we set the time limit to 10 hours. The long preset time aims to ensure that the resolution of the MIP model can obtain at least one feasible solution served as a comparison indicator with our ALNS algorithm, although in most cases it failed to achieve such a goal. To further evaluate the performance of our ALNS algorithm, we present the upper bounds found by CPLEX as well.

3.5.2.2 Parameter setting for the ALNS

The ALNS parameters were determined in preliminary experiments, since the ALNS algorithm is composed of several procedures and each procedure has its own parameters, parameter setting was tuned by concerning a tradeoff between solution quality and CPU time. The values used are ga-

thered in Table 3.2.

Observe that large size instances require more balanced probabilities (smaller selecting probability difference between $\{\gamma_1, \gamma_2, \gamma_3\}$) to provide the ALNS with more diverse initial solutions. The number of iterations without improvement δ before calling the meta-destroy mechanism must be increased on medium and large size instances, to give more time to the ALNS to explore its large neighborhoods. For the same reason, ρ , τ and θ are tuned in keeping with the size of instances.

TABLE 3.2: Parameter tuning according to instance size

| Symbol | Role | Small | Medium | Large |
|-------------|--|--------|--------|--------|
| $nruns$ | Number of runs | 10 | 10 | 10 |
| $nsegs$ | Segments per run | 10000 | 20000 | 50000 |
| $niters$ | Iterations per segment | 100 | 100 | 100 |
| γ_1 | SIH <i>Policy 1</i> probability | 0.15 | 0.15 | 0.20 |
| γ_2 | SIH <i>Policy 2</i> probability | 0.70 | 0.70 | 0.60 |
| γ_3 | SIH <i>Policy 3</i> probability | 0.15 | 0.15 | 0.20 |
| ϱ_1 | Operator score increment case 1 | 10 | 10 | 10 |
| ϱ_2 | Operator score increment case 2 | 5 | 5 | 5 |
| ϱ_3 | Operator score increment case 3 | 3 | 3 | 3 |
| ϱ_4 | Operator score increment case 4 | 1 | 1 | 1 |
| η | Score reaction factor | 0.8 | 0.8 | 0.8 |
| δ | Unfruitful iterations for meta-destroy | 100 | 150 | 200 |
| ρ | Removal fraction | 0.2 | 0.2 | 0.3 |
| τ | To set SA initial temperature | 0.3 | 0.4 | 0.4 |
| θ | SA cooling factor | 0.9995 | 0.9996 | 0.9997 |

3.5.3 Experimental results

Two key indicators were used to evaluate the performance of our ALNS algorithm :

i) The upper bound produced by CPLEX for the MIP model of a PDPTWPR instance input, which indicates the upper bound of the optimal

objective function value (profit).

ii) The best feasible solution of the MIP model found by CPLEX, which is marked as the lower bound of the optimal objective function value (profit).

For ease of reading, the abbreviations of the experiment indicators and corresponding definition are listed in Table 3.3.

TABLE 3.3: Abbreviation of experiment indicators and definition

| Abbreviation | Definition |
|------------------|--|
| UB_{MIP} | The upper bound of the MIP model obtained by CPLEX in a preset running time |
| LB_{MIP} | The best feasible objective value found by CPLEX solver in a preset running time |
| LB_{ALNS} | The best feasible objective value obtained by the ALNS after a preset number of iterations |
| Gap_{MIP} | The gap between LB_{MIP} and UB_{MIP} . It is calculated by : $\frac{UB_{MIP}-LB_{MIP}}{UB_{MIP}}$ |
| Gap_{ALNS} | The gap between LB_{ALNS} and UB_{MIP} . It is calculated by : $\frac{UB_{MIP}-LB_{ALNS}}{UB_{MIP}}$ |
| $Imp_{ALNS-MIP}$ | The improvement of LB_{ALNS} over LB_{MIP} . It is calculated by : $\frac{LB_{ALNS}-LB_{MIP}}{LB_{MIP}}$ |
| CPU_{ALNS} | The running time of the ALNS algorithm |
| CPU_{MIP} | CPU time for solving the MIP model by CPLEX |

Table 3.4 compares the performance of our ALNS algorithm and the CPLEX MIP solver on small size instances. For the instances with 10 requests, both solution approaches were able to solve the problem to optimality, but the ALNS algorithm consumed less CPU time than CPLEX. When the number of requests increases to 20, no proven optima were obtained, so we compare the near-optimal solutions of the two methods using the three above-mentioned indicators (Gap_{MIP} , Gap_{ALNS} and $Imp_{ALNS-MIP}$) and the running time of both. Observe that the ALNS algorithm found better objective value than CPLEX for 7 out of 9 instances and the average Gap_{ALNS} is only 6.11%. Furthermore, our ALNS algorithm supersedes CPLEX in terms of running time, the longest CPU time being only 23.2 seconds compared with the limit of 3600 seconds reached by CPLEX.

Table 3.5 gives in the same format the results for medium size instances. The improvement $Imp_{ALNS-MIP}$ increases quickly with the number of requests. For the group of instances with 30 requests, Gap_{ALNS} is on average

6.17% which maintains at the same level of instances with 20 requests, whereas Gap_{MIP} increases from 12.14% to 24.65%. For 40 requests instances, CPLEX failed to identify feasible solutions, in spite of a larger time limit of 2 hours. In contrast, our ALNS algorithm always returned good quality feasible solutions. For $n = 40$ and $n = 50$, the average Gap_{ALNS} is equal to 8.96% and 9.98% and $Imp_{ALNS-MIP}$ increases to 30.87% and 69.77%, respectively, excluding the cases for which CPLEX failed to obtain any feasible solution. In parallel, the running time of the ALNS grows naturally with instance size but still represents a small fraction of the CPU time consumed by CPLEX.

Table 3.6 summarizes the results for 100 requests instances. CPLEX achieved to find feasible solution for only 2 out of 9 instances. In most cases, CPLEX terminated because of lack of memory, so we did not try longer time limit. For this reason, we only report UB_{MIP} , LB_{ALNS} and Gap_{ALNS} for the remaining 7 instances to serve as a benchmark for future comparisons. Our algorithm produce an average 12.80% Gap_{ALNS} . Since the best upper bound might be reduced by more sophisticated techniques, the actual optimality gap is possibly less than Gap_{ALNS} .

Table 3.7 concludes the average value of Gap_{MIP} , Gap_{ALNS} , $Imp_{ALNS-MIP}$, CPU_{MIP} and CPU_{ALNS} by instance size, respectively.

TABLE 3.4: Computational results for small size instances

| Instances | UB_{MIP} | LB_{MIP}^a | LB_{ALNS}^a | $Gap_{MIP}(\%)^b$ | $Gap_{ALNS}(\%)^b$ | $Imp_{ALNS-MIP}(\%)$ | $CPU_{MIP}(s)$ | $CPU_{ALNS}(s)$ |
|--------------|------------|---------------|---------------|-------------------|--------------------|----------------------|----------------|-----------------|
| 10-5-5-50a | 965.2 | 965.2 | 965.2 | 0 | 0 | 0 | 47.4 | 4.2 |
| 10-5-5-50b | 1235.8 | 1235.8 | 1235.8 | 0 | 0 | 0 | 256.8 | 3.1 |
| 10-5-5-50c | 1415.0 | 1415.0 | 1415 | 0 | 0 | 0 | 123.3 | 3.2 |
| 10-3-7-50d | 1100.7 | 1100.7 | 1100.7 | 0 | 0 | 0 | 134.0 | 4.1 |
| 10-3-7-50e | 864.8 | 864.8 | 864.8 | 0 | 0 | 0 | 57.0 | 3.5 |
| 10-3-7-50f | 1467.6 | 1467.6 | 1467.6 | 0 | 0 | 0 | 52.2 | 3.0 |
| 10-7-3-50g | 1047.2 | 1047.2 | 1047.2 | 0 | 0 | 0 | 147.6 | 5.2 |
| 10-7-3-50h | 756.2 | 756.2 | 756.2 | 0 | 0 | 0 | 177.6 | 2.7 |
| 10-7-3-50i | 1226.4 | 1226.4 | 1226.4 | 0 | 0 | 0 | 101.2 | 4.5 |
| 20-10-10-50a | 4116.2 | 3765.1 | 3978.9 | 8.53 | 3.34 | 5.68 | 3600 | 14.4 |
| 20-10-10-50b | 3591.8 | 3123.3 | 3475 | 13.04 | 3.25 | 11.26 | 3600 | 12.2 |
| 20-10-10-50c | 1999.7 | 1863.6 | 1854.7 | 6.81 | 7.25 | -0.48 | 3600 | 9.3 |
| 20-5-15-50d | 3432 | 3007.4 | 3112.7 | 12.38 | 9.30 | 3.52 | 3600 | 11.0 |
| 20-5-15-50e | 3252.4 | 2766.6 | 3018.2 | 14.94 | 7.2 | 9.09 | 3600 | 10.3 |
| 20-5-15-50f | 2555.7 | 2365.0 | 2334.2 | 7.46 | 8.67 | -1.30 | 3600 | 23.2 |
| 20-15-5-50g | 4086.5 | 3461.3 | 3878.6 | 15.30 | 5.09 | 12.06 | 3600 | 14.1 |
| 20-15-5-50h | 3216.4 | 2682.0 | 2994.4 | 16.61 | 6.90 | 11.65 | 3600 | 17.0 |
| 20-15-5-50i | 4164.7 | 3574.3 | 4000.3 | 14.18 | 3.95 | 11.92 | 3600 | 10.3 |

^a The higher of the best feasible objective values found by the MIP (Column 3) and the ALNS (column 4) is indicated in boldface.

^b The better Gap between the upper bounds found by the MIP in a given running time and the best feasible solutions found by the ALNS/the best feasible solutions found by the MIP is emphasized in italics among the columns 5 and 6.

TABLE 3.5: Computational results for medium size instances

| Instances | UB_{MIP} | LB_{MIP}^a | LB_{ALNS}^a | $Gap_{MIP}(\%)^b$ | $Gap_{ALNS}(\%)^b$ | $Imp_{ALNS-MIP}(\%)$ | $CPU_{MIP}(s)$ | $CPU_{ALNS}(s)$ |
|--------------|------------|---------------|----------------|-------------------|--------------------|----------------------|----------------|-----------------|
| 30-15-15-50a | 9513.6 | 7235.3 | 9114.0 | 23.95 | 4.20 | 25.97 | 5400 | 42.1 |
| 30-15-15-50b | 11154.7 | 8963.8 | 10365.1 | 19.64 | 7.08 | 15.63 | 5400 | 30.0 |
| 30-15-15-50c | 10336.9 | 9456.4 | 9356.8 | 8.52 | 9.48 | -1.05 | 5400 | 35.7 |
| 30-10-20-50d | 12783.3 | 7177.7 | 11596.6 | 43.85 | 9.28 | 61.56 | 5400 | 56.3 |
| 30-10-20-50e | 11232.5 | 9632.4 | 10763.2 | 14.25 | 4.18 | 11.74 | 5400 | 46.0 |
| 30-10-20-50f | 8564.2 | 5864.3 | 7478 | 31.53 | 12.68 | 27.52 | 5400 | 60.5 |
| 30-20-10-50g | 10648.3 | 7364.0 | 10056.2 | 30.84 | 5.56 | 36.56 | 5400 | 41.1 |
| 30-20-10-50h | 10326.7 | 8264.7 | 10268.2 | 19.97 | 0.57 | 24.24 | 5400 | 47.8 |
| 30-20-10-50i | 8494.3 | 6002.0 | 8278.9 | 29.34 | 2.54 | 37.94 | 5400 | 42.0 |
| 40-20-20-50a | 14527.0 | 10023.6 | 12998.1 | 31.00 | 10.52 | 29.67 | 7200 | 72.1 |
| 40-20-20-50b | 15986.4 | 9552 | 14756.3 | 40.25 | 7.69 | 54.48 | 7200 | 102.0 |
| 40-20-20-50c | 15268.1 | 11752.1 | 13535.5 | 23.03 | 11.35 | 15.18 | 7200 | 80.6 |
| 40-15-25-50d | 12134.6 | 7531.8 | 11136.4 | 37.93 | 8.23 | 47.86 | 7200 | 118.0 |
| 40-15-25-50e | 10134.2 | 7963.9 | 9636.7 | 21.42 | 4.91 | 21.00 | 7200 | 75.6 |
| 40-15-25-50f | 10593.7 | - | 9616.0 | - | 9.23 | - | 7200 | 86.4 |
| 40-25-15-50g | 11667.4 | 8567.8 | 10589.3 | 26.57 | 9.24 | 23.59 | 7200 | 89.0 |
| 40-25-15-50h | 17868.5 | - | 16069.9 | - | 10.07 | - | 7200 | 84.3 |
| 40-25-15-50i | 13244.2 | 9654.0 | 12000.3 | 27.11 | 9.39 | 24.30 | 7200 | 100.3 |
| 50-25-25-50a | 26518.2 | 14421.2 | 24738.6 | 45.62 | 6.71 | 71.54 | 14400 | 361.0 |
| 50-25-25-50b | 21996.8 | - | 18991.0 | - | 13.66 | - | 14400 | 258.0 |
| 50-25-25-50c | 23644.1 | - | 22695.9 | - | 4.01 | - | 14400 | 247.5 |
| 50-20-30-50d | 22414.6 | - | 19983.6 | - | 10.85 | - | 14400 | 577.1 |
| 50-20-30-50e | 18649.9 | 10664.8 | 16119.2 | 42.82 | 13.57 | 51.14 | 14400 | 246.4 |
| 50-20-30-50f | 22378.0 | - | 20347.5 | - | 9.07 | - | 14400 | 416.0 |
| 50-30-20-55g | 19986.5 | 9894.5 | 18465.2 | 50.49 | 7.61 | 86.62 | 14400 | 365.9 |
| 50-30-20-50h | 23668.4 | - | 20004.1 | - | 15.48 | - | 14400 | 345.0 |
| 50-30-20-50i | 16986.0 | - | 15478.6 | - | 8.87 | - | 14400 | 466.8 |

^a The higher of the best feasible objective values found by the MIP (Column 3) and the ALNS (column 4) is indicated in boldface.

^b The better Gap between the upper bound found by the MIP in a given running time and the best feasible solutions found by the ALNS/the best feasible solutions found by the MIP is emphasized in italics among the columns 5 and 6.

TABLE 3.6: Computational results for large instances test

| Instances | UB_{MIP} | LB_{MIP}^a | LB_{ALNS}^a | $Gap_{MIP}(\%)^b$ | $Gap_{ALNS}(\%)^b$ | $Imp_{ALNS-MIP}(\%)$ | $CPU_{MIP}(s)$ | $CPU_{ALNS}(s)$ |
|----------------|------------|--------------|-----------------|-------------------|--------------------|----------------------|----------------|-----------------|
| 100-50-50-100a | 89554.8 | - | 74431.9 | - | 16.89 | - | 36000 | 741.2 |
| 100-50-50-100b | 94316.2 | 54549.7 | 85631.4 | 42.16 | 9.21 | 56.98 | 36000 | 766.8 |
| 100-50-50-100c | 127414.0 | - | 111717.1 | - | 12.32 | - | 36000 | 515.2 |
| 100-25-75-100d | 99874.7 | - | 86041.3 | - | 13.85 | - | 36000 | 1023.0 |
| 100-25-75-100e | 112084.5 | - | 96327.0 | - | 14.06 | - | 36000 | 985.1 |
| 100-25-75-100f | 96683.7 | 64493.4 | 82667.6 | 33.29 | 14.50 | 28.18 | 36000 | 602.0 |
| 100-75-25-100g | 81324.6 | - | 68543.2 | - | 15.72 | - | 36000 | 866.9 |
| 100-75-25-100h | 92333.1 | - | 84667.9 | - | 8.30 | - | 36000 | 711.4 |
| 100-75-25-100i | 13269.9 | - | 11898.0 | - | 10.34 | - | 36000 | 1176.2 |

^a The higher of the best feasible objective values found by the MIP (Column 3) and the ALNS (column 4) is indicated in boldface.

^b The better Gap between the upper bound found by the MIP in a given running time and the best feasible solutions found by the ALNS/the best feasible solutions found by the MIP is emphasized in italics among the columns 5 and 6.

TABLE 3.7: Statistical average value on the performance indicators

| Instance size | $Gap_{MIP}(\%)$ | $Gap_{ALNS}(\%)$ | $Imp_{ALNS-MIP}(\%)$ | $CPU_{MIP}(s)$ | $CPU_{ALNS}(s)$ |
|---------------|-----------------|------------------|----------------------|----------------|-----------------|
| $n = 10$ | 0 | 0 | 0 | 121.9 | 3.72 |
| $n = 20$ | 12.14 | 6.11 | 7.04 | 3600 | 13.53 |
| $n = 30$ | 24.65 | 6.17 | 26.68 | 5400 | 44.61 |
| $n = 40$ | 29.62 | 8.96 | 30.87 | 7200 | 89.81 |
| $n = 50$ | 46.31 | 9.98 | 69.77 | 14400 | 318.19 |
| $n = 100$ | 37.73 | 12.80 | 42.58 | 36000 | 820.87 |

3.5.4 Impact of the DAOB mechanism

In this subsection, we will analyze the gain of our proposed DAOB (see Section 3.4.3) from two perspectives. Firstly we present the proportion of feasible solutions as a function of instance size, with or without DAOB, in Fig. 3.8. The proportion of feasible solutions generated by the ALNS decreases sharply when the number of requests increases, whether the DAOB is activated or not, but clearly the DAOB technique looks effective since it ends with 70% of feasible solutions generated versus 50% when it is not activated.

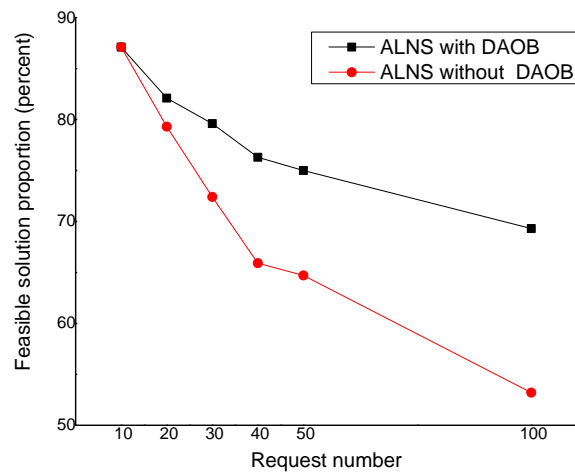


FIGURE 3.8: Proportion of feasible solutions generated with or without DAOB

Table 3.8 shows the average improvement of applying the DAOB for different size of instances. Here again, the DAOB leads to a better total profit (improve objective function value) on average.

TABLE 3.8: Average improvement of total profit with the DAOB

| Number of requests | Average improvement of total profit using the DAOB |
|--------------------|--|
| 10 | 0 |
| 20 | 5.7% |
| 30 | 7.0% |
| 40 | 10.3% |
| 50 | 9.8% |
| 100 | 14.6% |

3.6 Conclusion

This chapter introduces a new vehicle routing problem, the pickup and delivery problem with time windows, profits, and reserved requests

(PDPTWPR). The PDPTWPR is a key sub-problem of collaboration logistics (CL) in Less-than-truckload (LTL) transportation mode. To get a near optimal solution of the PDPTWPR under tight time windows and fleet size constraints, we develop specific techniques to improve the basic adaptive large neighborhood search (ALNS) method, such as the meta-destroy mechanism, the search organized in segments and the dynamic adjustment of operator behavior (DAOB). Moreover, eight tailored destroy/repair operators are designed to cope with the particularity of the PDPTWPR and a local search procedure based on six effective moves is added for further improvement. To the best of our knowledge, this is the first time that an ALNS is used for a pickup and delivery problem with profits.

To evaluate the performance of the ALNS heuristic applied to the PDPTWPR, a mixed-integer programming (MIP) model is formulated and solved by CPLEX in a pre-specified time limit. For small to medium size instances (up to 50 requests), the upper and lower bounds achieved by CPLEX are compared with the lower bounds obtained by the ALNS. The test results show that our heuristic is able to retrieve the proven optima found by CPLEX. In the cases without proven optima, the ALNS significantly outperforms CPLEX both in terms of solution quality and CPU time. For the large instances with 100 requests, even when CPLEX was not able to find a feasible solution in 10 hours, the ALNS was still able to generate a good feasible solution in a reasonable computation time.

Our future work will consider a variant of the PDPTWPR that has new characteristics such as a heterogeneous vehicle fleet, maximum tour duration, multiple vehicle depots, etc. The design of a fair post-collaboration profit reallocation scheme will also be addressed.

Stochastic Bid Generation Problem in Less-Than-Truckload Transportation

Contents

| | |
|---|----|
| 4.1 Introduction | 60 |
| 4.2 Problem description and mathematical model | 62 |
| 4.2.1 The stochastic MIQP model and its linearization | 68 |
| 4.2.2 Deterministic transformation of the probabilistic constraint | 69 |
| 4.2.3 Equivalent MIP model | 70 |
| 4.2.4 Synergy factor estimation | 72 |
| 4.3 GRASP × ILS hybrid | 75 |
| 4.3.1 Initial solution construction | 77 |
| 4.3.2 Local search operators | 77 |
| 4.3.3 Mutation | 79 |
| 4.4 Computational experiments | 79 |
| 4.4.1 Generation of instances | 79 |
| 4.4.2 Parameter setting | 81 |
| 4.4.3 Experimental results | 81 |

4.1 Introduction

This chapter addresses a stochastic bid generation problem (BGP) raised in combinatorial auctions (CA) for collaborative logistics (CL) in less-than-truckload (LTL) transportation, which combines request selection problem and routing problem for serving pickup and delivery requests with time windows. In this problem, multiple carriers form an alliance and exchange their transportation requests for the purpose of improving their profitability. Each carrier generates the best bid from requests for auction while still serving its own mandatory pickup and delivery requests with time windows.

Generally, in carrier collaboration, several carriers engage in the procedure of bid generation. Therefore, other carriers' behaviors of bidding should be considered when we try to solve the BGP from the point of view of one carrier. However, other carriers' behaviors of bidding are sometimes unpredictable, even unknown, because of business information protection issue. So, it is necessary to consider stochastic events (uncertainties) in the BGP processes to better simulate a free-market environment.

To the best of our knowledge, the study of the stochastic BGP in combinatorial transportation auctions or simultaneous transportation auctions is limited to truckload (TL) mode. The stochastic BGP in LTL transportation was never addressed before in the literature, although LTL transportation also plays an important role in nowadays logistics operations. So, in this chapter, we try to fill this gap by proposing a stochastic mixed-integer

quadratic programming (MIQP) model for BGP in LTL transportation and developing an effective heuristic algorithm. Motivated by the probabilistic optimization model in TL transportation proposed by [Triki et al. \(2014\)](#), we extend and apply it to LTL transportation. The hard MIQP model is then transformed into several independent pickup and delivery problems with time windows (PDPTW). A GRASP \times ILS algorithm is developed as the solution approach to the problem. Computational experiments on randomly generated instances show the effectiveness of the proposed GRASP \times ILS algorithm.

The contributions of this chapter are summarized as follows :

1. The stochastic BGP in LTL transportation is investigated for the first time.
2. The stochastic MIQP model of the problem is simplified and it is reduced to several independent PDPTW.
3. A reasonable technique is proposed to estimate the synergy factor among requests based on competitors' (other carriers') behaviors of bidding.
4. A GRASP \times ILS algorithm is developed to solve each PDPTW derived.

The remainder of this paper is organized as follows : Section 4.2 describes the problem studied in this paper and provides its MIQP model with a probabilistic constraint. Section 4.3 presents a simplified formulation of the mathematical model and the GRASP \times ILS algorithm. Computational experiments and analysis of their results are reported in Section 4.4. Finally, Section 4.5 concludes the chapter with perspectives for future research.

4.2 Problem description and mathematical model

The main purpose of this study is to build a BGP model which can help carriers to make decisions on generating and pricing bid(s) from a carrier's perspective in the context of LTL carrier collaboration, where multiple carriers collaborate with each other by exchanging part of their transportation requests for the purpose of improving their profitability.

A bid represents an offer to serve a bundle of requests with an 'ask price'. 'Ask price' indicates how much a carrier charges the auctioneer to serve the bundle of requests. Generally speaking, 'ask price' should be a positive value but not be superior to the sum of individual service costs of all requests in this bundle. 'Ask price' is a delicate issue in the BGP. On one hand, the more a carrier charges the auctioneer to serve a bundle of requests, the more the profit can be generated later. On the other hand, competitors (other carriers in coalition) may ask lower prices to serve the same bundle of requests, under such a circumstance, a losing bid gets nothing in return. Hereafter, we refer to the determination of 'ask price' as pricing problem.

Each transportation request is associated with a pair of pickup and delivery locations and time windows, quantity of goods, and price (paid by shippers/customers). Moreover, to adapt to realistic industry scenarios, we assume that each carrier has some reserved requests that must be served by itself and its other requests can be offered (outsourced) to other carriers in coalition. The outsourced requests are gathered and be referred to as requests for auction. The objective of the BGP for a carrier is to optimally select some requests for auction to bid for (generate the best bid) while still serving its reserved requests in the routing planning. Furthermore, other

carriers' behaviors of bidding should also be considered, because carriers are competitors during the bidding phase. In brief, three key issues must be taken into account simultaneously : 1) request selection problem : which requests to bid for 2) pricing problem : how to set the 'ask price' of a bundle of requests 3) routing problem : how to plan the routes for serving reserved requests and bidding requests.

For the sake of simplicity, we make the following assumptions (known information) for our stochastic BGP model :

- two types of requests : reserved requests and requests for auction
- service fee provided by shippers/customers for each request
- fleet size of the carrier (bidder) and its unique depot location
- other carriers' (competitors') depot locations
- probability of losing a bid

The vehicle routing problem (VRP) involves the design of a set of minimum cost routes for a fleet of vehicles which serve exactly once a set of customers with known demands. The PDPTW is a generalization of the VRP which is concerned with the construction of optimal routes to satisfy transportation requests, each requiring both pickup and delivery under capacity, time window and precedence constraints (Dumas, Desrosiers, & Soumis, 1991). The stochastic BGP studied in this paper is based on the PDPTW, and defined on a complete undirected graph $G = (N, E)$. The node-set is defined as $N = \{0, \dots, 2n + 1\}$, where n denotes the number of requests. Nodes 0 and $2n + 1$ represent the depot of the carrier, hosting a set $K = \{1, \dots, m\}$ of m identical vehicles of capacity Q . It is assumed that each vehicle route begins at node 0 and ends at node $2n + 1$. Each node i has a time window $[a_i, b_i]$ constraint, while each edge (i, j) in E has a travel cost c_{ij} and a travel time t_{ij} . The service times at node i is included in the

t_{ij} 's. Like in the VRPTW, a vehicle can wait at customer i if it arrives before a_i . The subset $P = \{1, \dots, n\}$ contains the pickup nodes of all requests, while $D = \{n + 1, \dots, 2n\}$ gathers delivery nodes. Request $i, i = 1, \dots, n$, is associated with a pickup node i , a delivery node $n + i$, a demand $d_i > 0$ and a price p_i . For the delivery node, we set $d_{n+i} = -d_i$. The set R of all requests includes the subset of carrier's reserved requests R_r and the subset of requests for auction R_s . The objective for the carrier is to maximize its total profit which is equal to the sum of collected payments of served requests minus the total cost of the routes. The load of a route before and after visiting each node cannot exceed vehicle capacity, the time window at each node must be respected, and the delivery node of each request must be visited after its corresponding pickup node, in the same route.

The problem is NP-hard in strong sense since it is a particular case of the PDPTW where R_s is empty. In our BGP, we consider other carriers' (competitors') behaviors of bidding in order to simulate a free market competition environment. Thus, the auction clearing price (lowest price) of each potential bid remains uncertain. In order to take into account this fact during the bidding phase, we involve the probabilistic constraint (4.2) (Triki et al., 2014) in our model. Let variables Y_b denote the auction clearing price of bid b that depends on the total price of requests in bundle and the synergy factor among them, and $\alpha \in [0, 1]$ represents probability of losing a bid.

In addition to the previous parameters, we need two additional relations to write the model more easily : $T_{ij} = b_j - a_i$ plays the role of a big-M constant in the time window constraints, while $Q_i = Q + d_i$ is used in the capacity constraints. We formulate the BGP as a MIQP model with a probabilistic constraint, and indices, sets, parameters and decision variables are introduced as follows :

Decision variables :

- x_{ij}^k , binary variable equals to 1 if and only if vehicle k travels directly through arc (i, j) ,
- y_i^k , binary variable equals to 1 if and only if request i is served by vehicle k ,
- z^b , binary variable equals to 1 if and only if bundle b is a generated bid,
- p^b , the 'ask price' of bundle b ,
- T_i^k , time at which vehicle k begins service at node i ,
- Q_i^k , load of vehicle k when leaving node i .

Sets :

- R_r , set of the carrier's (bidder's) reserved requests,
- R_s , set of requests for auction,
- R , set of all requests, $R = R_r \cup R_s$,
- B , set of all potential bids, indexed by b ,
- K , set of available vehicles, indexed by k .

Parameters :

- e_i^b , equals to 1 if and only if request i is in bid b , otherwise 0,
- p_{min}^b , small positive value indicating the minimum 'ask price' of bid b ,
- Y_b , random variable denoting the lowest price (auction clearing price) offered by competitors for bid b ,
- $\alpha \in [0, 1]$, probability of losing a bid,
- $p(R_r)$, total revenue obtained from serving the reserved requests,
- p_i , price of request i ,
- c_{ij} , transportation cost from node i to j , where $c_{ij} = c_{ji}$, and the triangle inequality holds,

66 Chapter 4. Stochastic bid generation problem in LTL transportation

- t_{ij} , travel time from node i to j ,
- d_i , quantity of request i ,
- $[a_i, b_i]$, time windows of node i ,
- Q , vehicle capacity.

Resulting model :

$$\max \sum_{b \in B} p^b \cdot z^b + p(R_r) - \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} \cdot x_{ij}^k \quad (4.1)$$

Subject to :

$$Pr(p^b \cdot z^b \leq Y_b) \geq 1 - \alpha \quad \forall b \in B \quad (4.2)$$

$$\sum_{b \in B} z^b \leq 1 \quad (4.3)$$

$$\sum_{j \in N, j \neq i} x_{ji}^k - \sum_{j \in N, j \neq i} x_{ij}^k = 0 \quad \forall i \in P \cup D, \forall k \in K \quad (4.4)$$

$$\sum_{j \in P, j \neq 0} x_{0j}^k = 1 \quad \forall k \in K \quad (4.5)$$

$$\sum_{i \in D, i \neq 2n+1} x_{i, 2n+1}^k = 1 \quad \forall k \in K \quad (4.6)$$

$$\sum_{k \in K} y_i^k = 1 \quad \forall i \in R_r \quad (4.7)$$

$$\sum_{k \in K} y_i^k = \sum_{b \in B} e_i^b \cdot z^b \quad \forall i \in R_s \quad (4.8)$$

$$\sum_{j \in N, j \neq i, 2n+1} x_{ij}^k = y_i^k \quad \forall i \in P, \forall k \in K \quad (4.9)$$

$$\sum_{j \in N, j \neq i, 0} x_{j, n+i}^k = y_i^k \quad \forall i \in P, \forall k \in K \quad (4.10)$$

$$T_i^k + t_{i, n+i} \leq T_{n+i}^k \quad \forall i \in P, \forall k \in K \quad (4.11)$$

$$T_j^k \geq T_i^k + t_{ij} \cdot x_{ij}^k - T_{ij} \cdot (1 - x_{ij}^k) \quad \forall i, j \in N, \forall k \in K \quad (4.12)$$

$$a_i \leq T_i^k \leq b_i \quad \forall i \in N, \forall k \in K \quad (4.13)$$

$$Q_j^k \geq Q_i^k + d_j - Q_j \cdot (1 - x_{ij}^k) \quad \forall i, j \in N, \forall k \in K \quad (4.14)$$

$$\max\{0, d_i\} \leq Q_i^k \leq \min\{Q, Q + d_i\} \quad \forall i \in N, \forall k \in K \quad (4.15)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in N, \forall k \in K \quad (4.16)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in R, \forall k \in K \quad (4.17)$$

$$z^b \in \{0, 1\} \quad \forall i \in R_s, \forall b \in B \quad (4.18)$$

$$p^b \in [p_{min}^b, \sum_{i \in b} p_i] \quad \forall b \in B \quad (4.19)$$

$$T_i^k \geq 0 \quad \forall i \in N, \forall k \in K \quad (4.20)$$

$$Q_i^k \geq 0 \quad \forall i \in N, \forall k \in K \quad (4.21)$$

The objective function (4.1) represents the expected profit, which equals to the difference between the sum of revenue and the total transportation cost. Constraints (4.2) impose a minimum probability for winning a tendered bid. Constraints (4.3) will force the model to generate at most one bid for each run, the most competitive one (one can solve again the same model if more bids are needed). Constraints (4.4) guarantee the flow balances at each node for each route. Constraints (4.5) and (4.6) indicate that each vehicle leaves the depot and returns to it. Constraints (4.7) impose that all reserved requests must be served. Constraints (4.8) represent the logical relationship between the routing variables and the binary bid variables. Constraints (4.9) and (4.10) ensure that if a request is served, there must be a vehicle leaving its pickup node and arriving at its paired delivery node. Time windows and precedence relations are respected via constraints (4.11) to (4.13). Constraints (4.14) and (4.15) involve vehicle capacity. Finally, constraints (4.16) to (4.21) define the variables.

Note that we enumerate all the potential bids for the set B from all the requests for auction, *i.e.*, the power-set of R_s in our model. The techniques to reduce the number of potential bids while still guaranteeing the same result as bidding on each element of the powerset of the set of requests for auction can be found in [Buer, 2014](#).

The proposed model is able to generate only one bid at a time, with its 'ask price' and the corresponding routing plan; Nevertheless, in order to adapt a wide range of bidding language mechanisms, *e.g.*, OR or XOR, the carrier can iteratively run the model to generate more bids as long as more bids exist from the remaining requests for auction.

4.2.1 The stochastic MIQP model and its linearization

Observe that the proposed BGP model is quadratic because of the product of non-negative variable p^b and binary variable z^b in objective function (4.1) and constraints (4.2). To make the model solvable by a commercial solver, *e.g.*, GUROBI or CPLEX (they can only solve convex quadratic programming, obviously, our model is non-convex), we linearize the stochastic MIQP model as follows :

Let $f^b = p^b \cdot z^b, \forall b \in B$, then we can rewrite the objective function (4.1) and constraints (4.2) as the objective function (4.22) and constraints (4.23) by adding new constraints (4.24) into the model :

$$\max \sum_{b \in B} f^b + p(R_r) - \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} \cdot x_{ij}^k \quad (4.22)$$

Subject to :

$$Pr(f^b \leq Y_b) \geq 1 - \alpha \quad \forall b \in B \quad (4.23)$$

$$p_{min}^b \cdot z^b \leq f^b \leq \sum_{i \in b} p_i \cdot z^b \quad \forall b \in B \quad (4.24)$$

The product of integer variable p^b and binary variable z^b is transformed into an integer variable f^b . So the stochastic MIQP model is converted into a stochastic mixed-integer programming (MIP) model. The value of f^b can be obtained by resolving the stochastic MIP model. That is, f^b equals to p^b with $z^b = 1$.

4.2.2 Deterministic transformation of the probabilistic constraint

To transform the probabilistic constraint (4.2) into a deterministic constraint, we assume that the price of each request for auction follows a normal distribution. A Gaussian distribution reflects some unpredictable characteristics of a request in real business situations, *e.g.*, expedited shipping, dangerous goods delivery, fragile items transport, etc.

More precisely, consider a bid b in which each request i has a price p_i paid by its shipper/customer. Assume that the prices of transportation requests are independent and that each p_i follows a normal distribution with mean μ_i and variance σ_i^2 , where μ_i and σ_i^2 are estimated from the distance between pickup and delivery nodes, the load d_i , and the record of past order history : $p_i \equiv N(\mu_i, \sigma_i^2)$.

Setting the ask price of a bid must consider complementarity among transportation requests in the bid and the bidding strategies of other car-

riers. The complementarity among requests in a bid b can be measured by a synergy factor S_b between 0 and 1. That is, the auction clearing price of this bid is estimated as $Y_b = S_b \cdot \sum_{i \in b} p_i, \forall b \in B$. The smaller the synergy factor, the higher the complementarity. (Triki et al., 2014) pointed out that one of challenges for the BGP is to estimate the synergy factor S_b properly. This key-issue is addressed in subsection 4.2.4.

Since Y^b involves a sum of independent Gaussian parameters, it follows that $Y_b \equiv N(\mu'_b, \sigma_b'^2)$, where $\mu'_b = S_b \cdot \sum_{i \in b} \mu_i$, and $\sigma_b'^2 = S_b \cdot \sum_{i \in b} \sigma_i^2$. Consequently, $Y_b \equiv N(S_b \cdot \sum_{i \in b} \mu_i, S_b \cdot \sum_{i \in b} \sigma_i^2)$ and we can transform the probabilistic constraints (4.23) as follows :

$$f^b \leq S_b \cdot \left[\sum_{i \in b} \mu_i + \Phi^{-1}(\alpha) \cdot S_b \cdot \sum_{i \in b} \sigma_i^2 \right], \forall b \in B \quad (4.25)$$

where Φ^{-1} represents the inverse function of the cumulative distribution function for the standard normal distribution. Recall that α denotes the probability of losing a bid.

4.2.3 Equivalent MIP model

After the linearization manipulation and deterministic transformation of the probabilistic constraint upon the stochastic MIQP model, it results a new MIP model which is equivalent to the original one.

$$\max \sum_{b \in B} f^b + p(R_r) - \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} \cdot x_{ij}^k \quad (4.26)$$

Subject to :

$$\sum_{b \in B} z^b \leq 1 \quad (4.27)$$

$$f^b \leq S_b \cdot \left[\sum_{i \in b} \mu_i + \Phi^{-1}(\alpha) \cdot S_b \cdot \sum_{i \in b} \sigma_i^2 \right] \quad \forall b \in B \quad (4.28)$$

$$p_{min}^b \cdot z^b \leq f^b \leq \sum_{i \in b} p_i \cdot z^b \quad \forall b \in B \quad (4.29)$$

$$\sum_{j \in N, j \neq i} x_{ji}^k - \sum_{j \in N, j \neq i} x_{ij}^k = 0 \quad \forall i \in P \cup D, \forall k \in K \quad (4.30)$$

$$\sum_{j \in P, j \neq 0} x_{0j}^k = 1 \quad \forall k \in K \quad (4.31)$$

$$\sum_{i \in D, i \neq 2n+1} x_{i,2n+1}^k = 1 \quad \forall k \in K \quad (4.32)$$

$$\sum_{k \in K} y_i^k = 1 \quad \forall i \in R_r \quad (4.33)$$

$$\sum_{k \in K} y_i^k = \sum_{b \in B} e_i^b \cdot z^b \quad \forall i \in R_s \quad (4.34)$$

$$\sum_{j \in N, j \neq i, 2n+1} x_{ij}^k = y_i^k \quad \forall i \in P, \forall k \in K \quad (4.35)$$

$$\sum_{j \in N, j \neq i, 0} x_{j,n+i}^k = y_i^k \quad \forall i \in P, \forall k \in K \quad (4.36)$$

$$T_i^k + t_{i,n+i} \leq T_{n+i}^k \quad \forall i \in P, \forall k \in K \quad (4.37)$$

$$T_j^k \geq T_i^k + t_{ij} \cdot x_{ij}^k - T_{ij} \cdot (1 - x_{ij}^k) \quad \forall i, j \in N, \forall k \in K \quad (4.38)$$

$$a_i \leq T_i^k \leq b_i \quad \forall i \in N, \forall k \in K \quad (4.39)$$

$$Q_j^k \geq Q_i^k + d_j - Q_j \cdot (1 - x_{ij}^k) \quad \forall i, j \in N, \forall k \in K \quad (4.40)$$

$$\max\{0, d_i\} \leq Q_i^k \leq \min\{Q, Q + d_i\} \quad \forall i \in N, \forall k \in K \quad (4.41)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in N, \forall k \in K \quad (4.42)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in R, \forall k \in K \quad (4.43)$$

$$z^b \in \{0, 1\} \quad \forall i \in R_s, \forall b \in B \quad (4.44)$$

$$p^b \in [p_{min}^b, \sum_{i \in b} p_i] \quad \forall b \in B \quad (4.45)$$

$$T_i^k \geq 0 \quad \forall i \in N, \forall k \in K \quad (4.46)$$

$$Q_i^k \geq 0 \quad \forall i \in N, \forall k \in K \quad (4.47)$$

4.2.4 Synergy factor estimation

In this subsection, we discuss how to estimate the synergy factor among requests in a bid. Note that even for the TL transportation, the issue of bundle synergy computation was rarely addressed in the literature. A rigorous study of synergy among requests needs to involve the exact modeling technique and the evaluation of interactions among numerous dependent random variables, which is not a trivial task.

However, it is necessary to estimate the synergy factor of a bid since we have to predict other carriers' lowest offers to serve this bid in order to win the bid (see constraint(4.2)). This synergy information makes the carrier who is solving the stochastic BGP to offer an 'ask price' lower than those of its competitors in order to increase the probability of winning the bid. So contrary to evaluating the synergy from requests' interactions (An, Elmaghraby, & Keskinocak, 2005; Wang & Xia, 2005; Lee et al., 2007; Chang, 2009), we propose a new method to estimate the synergy among requests taking account of the competitions from all other carriers.

Let denote the set of other carriers by W . For each of other carrier $w \in W$, we first estimate the synergy factor of a bid b with respect to it by the procedure described as follows :

1. calculate the optimal cost of the carrier w to serve bid b without considering its reserved requests, denote this cost as $C_{optimal}^{w,b}$
2. calculate the total cost of the carrier w to serve each request in bid b independently, denote it as $\sum_{i \in b} C_i^{w,b}$
3. the synergy factor of bid b with respect to the carrier w is estimated by the formula : $S^{w,b} = C_{optimal}^{w,b} / \sum_{i \in b} C_i^{w,b}$

Note that the cost matrix used in the above calculation is taken as that of

the carrier who solves the BGP and we only take into account each carrier's depot without considering its reserved requests for the synergy estimation. Because reserved requests are private business information in most cases, carriers disclose them unwillingly even to partners in realistic situations, whereas their vehicle depots are normally the information known by the public.

The value of $S^{w,b} \in (0, 1)$ reflects the degree of complementarity among requests in bundle b for carrier w , a smaller value of $S^{w,b}$ implies a more complementary synergy of requests in the bid for this carrier. In other words, this carrier would ask a relative lower price to serve the bid. For the carrier who is solving its BGP to determine a bid to submit in order to estimate S_b for each bid b , it first estimates the synergy of the bid with respect to each of other carriers in coalition by the procedure presented above, and then takes the smallest value of the synergies as S_b . This is because for the carrier, in order to win a bid, its 'ask price' should beat the 'ask prices' of the bid offered by all other carriers.

To make more clearly the synergy estimation procedure, a simple example is used to illustrate the evaluation of the synergy factor of a bid o with two requests 1, 2 competed by three carrier a, b, c as given in Table 4.1. It is assumed that each carrier has only one vehicle with capacity 30. The optimal route to serve the bid o for carrier a is $D_a \rightarrow P_2 \rightarrow D_2 \rightarrow P_1 \rightarrow D_1 \rightarrow D_a \Rightarrow C_{optimal}^{a,o} = 22.07$. If carrier a serves the two requests independently, the routes will be $D_a \rightarrow P_1 \rightarrow D_1 \rightarrow D_a$ with cost 12.07, and $D_a \rightarrow P_2 \rightarrow D_2 \rightarrow D_a$ with cost 12.07. Then we can have the total cost $\sum_{i \in o} C_i^{a,o} = 24.14$. The synergy factor $S^{a,o}$ can thus be estimated as $C_{optimal}^{a,o} / \sum_{i \in o} C_i^{a,o} = 0.9143$ (Figure 4.1). The value 0.9143 means carrier a may achieve around 8.57% cost saving to serve this bid o from the eco-

74 Chapter 4. Stochastic bid generation problem in LTL transportation

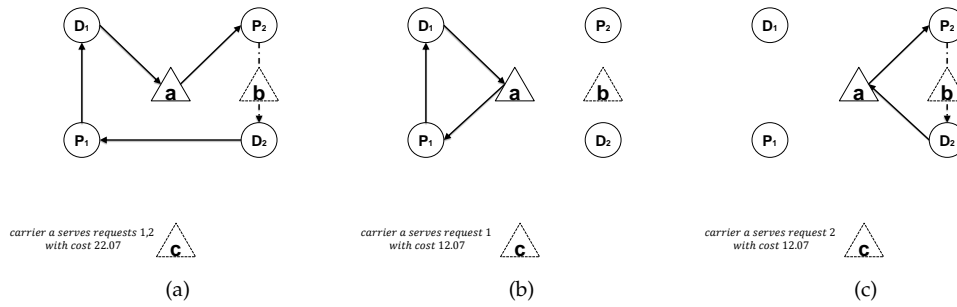


FIGURE 4.1: Example of cost calculation for synergy factor estimation for carrier a .

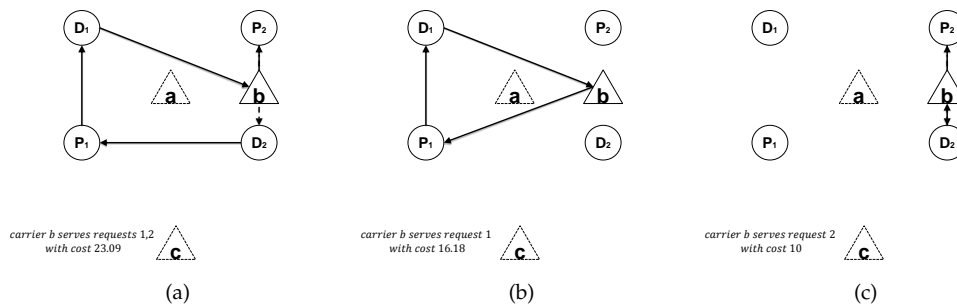


FIGURE 4.2: Example of cost calculation for synergy factor estimation for carrier b .

nomies of scope compared with serving them separately. Similarly, we can have $S^{b,o} = 0.8819$ (Figure 4.2), $S^{c,o} = 0.8517$ (Figure 4.3). The smallest value among $S^{a,o}$, $S^{b,o}$, and $S^{c,o}$ is 0.8517, so the synergy factor of bid o is estimated as 0.8517.

TABLE 4.1: Example data for synergy factor estimation

| Node index | Coordinates | Time window | Quantity |
|--------------------------------------|-------------|-------------|----------|
| Depot of carrier a (D_a) | (7.5,7.5) | (0,5000) | / |
| Depot of carrier b (D_b) | (10,7.5) | (0,5000) | / |
| Depot of carrier c (D_c) | (7.5,0) | (0,5000) | / |
| Pickup node of request 1 (P_1) | (5,5) | (2406,2945) | 15 |
| Delivery node of request 1 (D_1) | (5,10) | (3537,4283) | -15 |
| Pickup node of request 2 (P_2) | (10,10) | (1145,2199) | 12 |
| Delivery node of request 2 (D_2) | (10,5) | (1453,2702) | -12 |

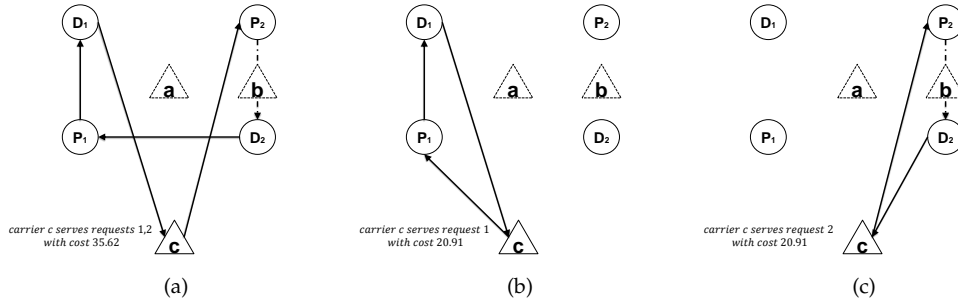


FIGURE 4.3: Example of cost calculation for synergy factor estimation for carrier c .

4.3 GRASP \times ILS hybrid

In Subsection 4.2.3, we present the MIP model which is equivalent to the stochastic BGP in LTL carrier collaboration. Since existing commercial solvers are not powerful enough to solve the PDPTW of large size in a reasonable time, and the problem studied in this paper is even harder than the PDPTW. The challenge imposes us to develop a fast and effective heuristic algorithm.

Observe that we can enumerate all potential bids $b \in B$ to simplify the MIP model due to the XOR bidding language (constraint (4.3)), only one bid is generated after each run of the MIP model. Since for each bid generated with $z_b = 1$, $f^b = \min\{\sum_{i \in b} p_i, S_b \cdot [\sum_{i \in b} \mu_i + \Phi^{-1}(\alpha) \cdot S_b \cdot \sum_{i \in b} \sigma_i^2]\}$ maximizes the objective function. Hence, the problem is transformed into m sub-problems, where m is the number of potential bids. Each sub-problem is equivalent to the PDPTW. Because the carrier is obliged to serve all the reserved requests and the requests in the selected bid with a fixed total revenue. Thus, the minimization of total traveled distance is the objective of each sub-problem.

A large number of potential bids (increases exponentially with the number of requests for auction) motivates us to implement a light, simple and

also powerful heuristic as the solution approach. GRASP \times ILS is a pertinent heuristic algorithm for the task because of its simple structure, low time complexity and customized local search components.

The general structure of our algorithm is sketched in Algorithm 5 and its components are detailed in the following subsections. S and $f(S)$ represent a solution and its profit. $RSIH$ denotes a randomized sequential insertion heuristic to generate initial solutions (Subsection 4.3.1) and $Mutate$ is a random mutation procedure like that in genetic algorithms (Subsection 4.3.3). LS is a local search procedure (Subsection 4.3.2). S^* and $f^* = f(S^*)$ respectively denote the best solution found by the GRASP \times ILS and its profit. For the parameters, np denotes the number of phases (each phase generates a local optimum), ni indicates the number of iterations per phase (number of attempts to produce better local optima). The total number of calls to the local search is $ncls = np \times ni$ (Prins, 2009).

Algorithm 5 – GRASP \times ILS hybrid

```

1: initialize the random number generator
2:  $f^* \leftarrow -\infty$ 
3: for  $i \leftarrow 1$  to  $np$  do
4:    $RSIH(S)$  (Subsection 4.3.1)
5:    $LS(S)$  (Subsection 4.3.2)
6:   for  $j \leftarrow 1$  to  $ni$  do
7:      $S' \leftarrow S$ 
8:      $Mutate(S')$  (Subsection 4.3.3)
9:      $LS(S')$ 
10:    if  $f(S) < f(S')$  then
11:       $S \leftarrow S'$ 
12:    end if
13:    if  $f^* < f(S)$  then
14:       $f^* \leftarrow f(S)$ 
15:       $S^* \leftarrow S$ 
16:    end if
17:  end for
18: end for

```

4.3.1 Initial solution construction

A randomized sequential insertion heuristic (RSIH) is developed to construct a random initial solution before a local search procedure, see Algorithm 6. RSIH builds routes one by one. Carrier's reserved requests and requests in the selected bid are combined together and sorted in decreasing order of their distances to depot. In each step, the l^{st} ($l \in q, l \in N$) farthest request is picked to insert into the current routing plan, here a random factor q (Subsection 4.4.2) is involved to arouse the effect of multi-start solution construction. Then all potential insertion places for request i are checked and memorized over the current routing plan. If no feasible insertion place exists, a new route will be created and a new step of picking next request to insert is invoked. Otherwise, the picked request i is inserted into its best insertion place. In this heuristic, the best insertion place corresponds to the smallest detour in distance, where the detour distance is $d_{ji} + d_{ik} - d_{jk}$ for the insertion of node i between j and k . Here, the detour distance of request i is calculated based on the sum of its pickup node i and its pairwise delivery node $n+i$. The procedure will repeat until all requests are inserted.

4.3.2 Local search operators

We adopt in our GRASP \times ILS algorithm four local search operators : intra/inter-route relocations and intra/inter-route exchanges of requests. This article describes also a 2-opt move but time windows make it infeasible in most cases. So, we discard it to reduce the computational burden. At the beginning of each iteration of the local search, a random ordering of the four neighborhoods is selected. The neighborhoods are browsed in this

Algorithm 6 – Randomized sequential insertion heuristic (RSIH)

```
1: merge reserved requests and bidding requests in array  $L$ 
2: sort  $L$  by descending distance to depot
3:  $r \leftarrow 1$ 
4: while  $|L| > 0$  do
5:   choose a random integer number  $l$  in the interval  $[0, q]$ 
6:   pick request :  $i = L[l]$ 
7:   if there is no feasible insertion place for request  $i$  then
8:      $r \leftarrow r + 1$ 
9:     continue
10:  else
11:    insert request  $i$  to its best insertion place
12:    remove request  $i$  from  $L$ 
13:  end if
14: end while
```

order and the first improving move detected is executed. The local search stops after a maximum number of iterations ni or no improvement move can be found.

Algorithm 7 – Local search procedure (LS)

```
1:  $\delta \leftarrow 0$  (number of iterations)
2: repeat
3:   define a random ordering of the four neighborhoods
4:   search for the first improving move  $S \rightarrow S'$  in this order
5:   if one improving move is found then
6:      $S \leftarrow S'$ 
7:   else
8:     break
9:   end if
10:   $\delta \leftarrow \delta + 1$ 
11: until  $\delta = ni$ 
```

4.3.3 Mutation

The mutation in the GRASP \times ILS algorithm swaps two distinct node i and j without violating any capacity, time window and precedence constraints. p successive swaps are executed in each mutation. The level p is set to p_{min} at the beginning of each phase or at each time the best solution is improved. It is increased by one each time the mutation following local search returns a no-improved solution, but without exceeding a maximum value p_{max} . The interval of level p is detailed in Subsection 4.4.2.

4.4 Computational experiments

The numerical experiments were conducted on a desktop equipped with an Intel Core i7-2600 3.40 gigahertz processor, 8 GB gigabyte of RAM, and Windows 7 Professional (64 bits) service pack 1. The GRASP \times ILS algorithm and the instance generator described in the sequel were both implemented by using Python programming language (Interpreter : Cpython 0.24.1, IDE : JetBrains PyCharm 2016.2.3). The MIP model for the stochastic BGP was built and solved by using the GUROBI MIP solver (version 6.5.2) integrated in its API for Python with the following parameter settings : MIPFocus = 1, Quad = 1 , NumericFocus = 2 , Threads = 4 (see GUROBI 6.5.2 Reference Manual).

4.4.1 Generation of instances

Since the MIQP model for carrier collaboration in LTL transportation is developed for the first time, and no benchmark instance is publicly available in the literature. So we decided to construct a specific instance gene-

rator.

The instances of this study are generated based on the benchmark instances in [Ropke et Cordeau \(2009\)](#), available at <http://www.diku.dk/~sropke/>.

We directly copy the instances with the coordinates of each node, the demand and time windows of each request. Each instance is given a label of *source code - number of requests - number of bids - number of vehicles - number of competitors*. Consider instance AA30-7-7-3-4 as an example. First code 'AA30' indicates that this instance is modified from the instance 'AA30', second code means this instance includes 7 requests in total (including reserved requests and requests for auction). Only the first 7 requests appearing in AA30 are considered in instance AA30-7-7-3-4. Third code shows there are 7 potentials bids, and fourth code indicates the size of fleet is 3. Last code represents 4 competitors are taken into account.

We consider two cases to generate the average price of each request μ . One takes into account both distance factor and quantity factor, the other one only considers distance factor. For the first case, $\mu = \rho \times distance \times load$, and second case $\mu = \rho \times distance$, where *distance* represents the average distance between the pickup node and the delivery node of each request in bid, and *load* represents the average quantity of demands of each request in bid. A random factor $\rho \in (0, 1)$ is used to simulate the record of the past order history. For the variance σ^2 , we set it as ten percent of μ . For the generation of the set of bids B , we enumerate all the potential bids as described in Subsection 4.2. p_{min}^b is set to be 10^{-5} , and α , the probability threshold, is in the interval $(0, 0.1]$, which can be found in the instance files. The fleet size is adjusted in accordance with the proportion of the number of requests extracted from the original instance. Finally, since some original

instances have multiple depots, we select one of the depots arbitrarily in our case.

4.4.2 Parameter setting

The GRASP \times ILS algorithm is controlled by relatively few parameters. Their values were determined by preliminary experiments in order to achieve a good trade-off between solution quality and CPU time. As shown in Table 4.2, the parameters' values depend on instance size. For ease of reading, we recall the meaning of p and q : p indicates the interval for the mutation level (Subsection 4.3.3), and q represents the interval for the randomized initial solution construction (Subsection 4.3.1).

TABLE 4.2: Parameter setting of the GRASP \times ILS according to instance size

| Symbol | Role | Small | Medium |
|--------|---|--------|--------|
| np | number of phases (call for multi-start) | 3 | 10 |
| ni | number of iterations (call for local search) | 100 | 500 |
| p | mutation level | [1, 1] | [1, 2] |
| q | randomized level of initial solution construction | [0, 1] | [0, 1] |

4.4.3 Experimental results

Since no benchmark of the BGP in LTL exists and the instances data are generated by our-self, one way to assess the performance of the GRASP \times ILS is to compare it with the exact solution. In this paper, we use GUROBI, a top commercial MIP solver to solve the simplified model of the stochastic BGP (constraint (4.26) to constraint (4.47)). However, solving exactly the model is only limited to instances with few requests for auction and few potentials bids.

For the GRASP \times ILS, we report the best feasible solution of each instance, denoted by LB , found by the algorithm when the stopping criteria is reached. The upper bound (UB), the lower bound (LB), and the Gap between UB and LB obtained by the GUROBI MIP solver are recorded. The running time is reported when the optimal solution is found or a prespecified time is reached instead. A bold face indicates a better feasible solution obtained from the GRASP \times ILS or GUROBI, and an asterisk means the solution is proven to be optimal by the GUROBI MIP solver. At last, the relative improvement of the GRASP \times ILS over the GUROBI MIP solver is also reported, denoted as Imp .

Table 4.3 and Table 4.4 compare the performance of the GRASP \times ILS algorithm and GUROBI on small size instances. For the instances with up to 10 reserved requests and requests for auction, and 31 potential bids. Both approaches are able to solve the model to optimality, but the GRASP \times ILS algorithm consumes less CPU time than GUROBI.

Table 4.5 and Table 4.6 give in the same format the results for medium size instances for two cases of price generation methods. Observe that the GUROBI MIP solver begins to exhibit poor performance with the increase of problem size. By contrast, the performance of GRASP \times ILS still remains stable with only a fraction of running time of the GUROBI MIP solver.

TABLE 4.3: GRASP \times ILS versus GUROBI MIP solver on small size instances ($\mu = \rho \times distance \times load$)

| Instance | GRASP \times ILS | | GUROBI | | | | Imp(%) |
|----------------|--------------------|-----------|--------|------------------|-----------|-----------|--------|
| | LB | $Time(s)$ | UB | LB | $Gap(\%)$ | $Time(s)$ | |
| AA30-7-7-3-4 | 1637.10 | 0.42 | - | * 1637.10 | - | 0.95 | 0 |
| BB30-7-7-3-4 | 1521.71 | 0.46 | - | * 1521.71 | - | 0.82 | 0 |
| AA30-10-31-3-3 | 2394.28 | 8.23 | - | * 2394.28 | - | 24.97 | 0 |
| BB30-10-31-3-3 | 2395.30 | 7.57 | - | * 2395.30 | - | 35.77 | 0 |
| Average | - | 4.17 | - | - | - | 15.63 | 0 |

TABLE 4.4: GRASP \times ILS versus GUROBI MIP solver on small size instances ($\mu = \rho \times distance$)

| Instance | GRASP \times ILS | | GUROBI | | | | Imp(%) |
|----------------|--------------------|----------------|-----------|-----------------|---------------|----------------|--------|
| | <i>LB</i> | <i>Time(s)</i> | <i>UB</i> | <i>LB</i> | <i>Gap(%)</i> | <i>Time(s)</i> | |
| AA30-7-7-3-4 | 344.95 | 0.43 | - | * 344.95 | - | 0.91 | 0 |
| BB30-7-7-3-4 | 229.3 | 0.46 | - | * 229.3 | - | 0.61 | 0 |
| AA30-10-31-3-3 | 547.95 | 7.94 | - | * 547.95 | - | 33.11 | 0 |
| BB30-10-31-3-3 | 326.41 | 8.21 | - | * 326.41 | - | 33.07 | 0 |
| Average | - | 4.26 | - | - | - | 16.93 | 0 |

TABLE 4.5: GRASP \times ILS versus GUROBI MIP solver on medium size instances ($\mu = \rho \times distance \times load$)

| Instance | GRASP \times ILS | | GUROBI | | | | Imp(%) |
|-------------------|--------------------|----------------|-----------|------------------|---------------|----------------|--------|
| | <i>LB</i> | <i>Time(s)</i> | <i>UB</i> | <i>LB</i> | <i>Gap(%)</i> | <i>Time(s)</i> | |
| AA30-15-31-5-5 | 3196.51 | 183.66 | 3356.62 | 3189.06 | 4.99 | 3600.00 | 0.23 |
| BB30-15-31-5-5 | 3184.84 | 204.84 | - | * 3184.84 | - | 2975.92 | 0 |
| CC30-15-31-5-5 | 2130.23 | 196.70 | 2340.14 | 2090.87 | 10.65 | 3600.00 | 1.85 |
| DD30-15-31-5-5 | 2257.83 | 201.36 | 2526.74 | 2222.54 | 12.04 | 3600.00 | 1.56 |
| XX30-16-15-4-4 | 2463.30 | 166.74 | 2523.02 | 2394.56 | 5.09 | 3600.00 | 2.79 |
| YY30-16-15-4-4 | 2217.47 | 179.62 | 2508.35 | 2070.87 | 17.44 | 3600.00 | 6.61 |
| AA30-25-31-7-6 | 4109.48 | 328.40 | 4465.63 | 3938.95 | 11.79 | 7200.00 | 4.15 |
| BB30-24-63-7-8 | 4199.16 | 547.66 | 4572.49 | 3809.92 | 16.68 | 7200.00 | 9.30 |
| CC30-25-31-8-8 | 1704.05 | 295.62 | 2264.87 | 1476.12 | 34.83 | 7200.00 | 13.38 |
| DD30-25-31-8-8 | 1746.57 | 333.85 | 2315.68 | 1174.08 | 49.30 | 7200.00 | 32.78 |
| XX30-23-127-8-7 | 2006.80 | 1017.90 | 2344.08 | 1959.49 | 16.41 | 7200.00 | 2.36 |
| YY30-24-63-5-7 | 1848.30 | 603.42 | 2541.98 | 1818.99 | 28.44 | 7200.00 | 1.56 |
| AA40-32-255-8-10 | 524.11 | 2577.11 | 1046.11 | 232.28 | 77.80 | 14400.00 | 55.68 |
| BB40-32-255-10-10 | 2454.13 | 2396.73 | 3226.31 | 2157.86 | 33.17 | 14400.00 | 12.07 |
| CC40-35-31-10-10 | 2123.35 | 493.46 | 3405.85 | 2005.07 | 41.13 | 14400.00 | 5.57 |
| DD40-33-127-12-5 | 1372.28 | 1574.92 | 2413.87 | 531.46 | 77.98 | 14400.00 | 61.27 |
| XX40-33-127-11-5 | 1844.25 | 1321.38 | 2464.55 | 1097.49 | 55.47 | 14400.00 | 40.49 |
| YY40-32-255-10-6 | 1366.24 | 2782.83 | 2650.36 | 1072.54 | 59.53 | 14400.00 | 21.50 |
| Average | - | 855.90 | - | - | 30.71 | 8365.33 | 15.18 |

TABLE 4.6: GRASP \times ILS versus GUROBI MIP solver on medium size instances ($\mu = \rho \times distance$)

| Instance | GRASP \times ILS | | GUROBI | | | | Imp(%) |
|-------------------|--------------------|----------------|-----------|-----------|---------------|----------------|--------|
| | <i>LB</i> | <i>Time(s)</i> | <i>UB</i> | <i>LB</i> | <i>Gap(%)</i> | <i>Time(s)</i> | |
| AA30-15-31-5-5 | 609.7 | 195.37 | 729.38 | 606.34 | 16.87 | 3600.00 | 0.55 |
| BB30-15-31-5-5 | 639.25 | 184.66 | 713.91 | 634.3 | 11.15 | 3600.00 | 0.77 |
| CC30-15-31-5-5 | 805.22 | 184.72 | 978.68 | 790.68 | 19.21 | 3600.00 | 1.81 |
| DD30-15-31-5-5 | 611.18 | 174.11 | 861.68 | 575.34 | 33.23 | 3600.00 | 5.86 |
| XX30-16-15-4-4 | 783.73 | 202.84 | 825.23 | 774.34 | 6.17 | 3600.00 | 1.20 |
| YY30-16-15-4-4 | 754.47 | 179.36 | 986.27 | 669.13 | 32.16 | 3600.00 | 11.31 |
| AA30-25-31-7-6 | 1022.65 | 395.74 | 1355.13 | 976.42 | 27.95 | 7200.00 | 4.52 |
| BB30-24-63-7-8 | 996.57 | 641.13 | 1423.68 | 931.93 | 34.54 | 7200.00 | 6.49 |
| CC30-25-31-8-8 | 259.52 | 303.55 | 870.95 | 54.25 | 93.77 | 7200.00 | 79.10 |
| DD30-25-31-8-8 | 642.39 | 323.41 | 1254.04 | 429.07 | 65.78 | 7200.00 | 33.21 |
| XX30-23-127-8-7 | 1189.23 | 1064.88 | 1563.74 | 871.21 | 44.29 | 7200.00 | 26.74 |
| YY30-24-63-5-7 | 1163.88 | 642.74 | 1674.09 | 998.48 | 40.36 | 7200.00 | 14.21 |
| AA40-32-255-8-10 | 549 | 2471.23 | 1107.76 | 277.21 | 74.98 | 14400.00 | 49.51 |
| BB40-32-255-10-10 | 547.98 | 2330.43 | 1266.49 | 201.77 | 84.07 | 14400.00 | 63.18 |
| CC40-35-31-10-10 | 262.27 | 570.11 | 917.66 | 46.24 | 94.96 | 14400.00 | 82.37 |
| DD40-33-127-12-5 | 594.37 | 1530.6 | 841.02 | 313.54 | 62.72 | 14400.00 | 47.25 |
| XX40-33-127-11-5 | 790.84 | 1410.28 | 1286.59 | 107.95 | 91.61 | 14400.00 | 86.35 |
| YY40-32-255-10-6 | 616.63 | 2454.71 | 973.27 | 424.21 | 56.41 | 14400.00 | 31.21 |
| Average | - | 847.77 | - | - | 49.46 | 8400 | 30.31 |

4.5 Conclusions

The BGP is a key sub-problem in combinatorial auctions (CAs) for collaborative logistics (CL). Previous studies were limited to develop models and algorithms for the bid generation problem (BGP) in truckload (TL) transportation. In this paper, we extend the BGP to carrier collaboration in less-than-truckload (LTL) transportation. A mixed integer quadratic programming (MIQP) model with a probabilistic constraint is formulated, which simultaneously integrates request selection problem, pricing problem and routing problem. To tackle the intractable MIQP model, deterministic transformation and linearization technique are used to transform it into a mixed integer programming (MIP) model. The MIP model can be decomposed into several independent sub-problems, and each of them is

a pickup and delivery problem with time windows (PDPTW). For solving the PDPTW effectively and efficiently, a GRASP \times ILS algorithm is developed as the solution approach. In the algorithm, a randomized sequential initial solution heuristic and four customized local search operators are designed as its components. The performance of the GRASP \times ILS heuristic applied to the stochastic BGP is compared with a top commercial MIP solver, GUROBI. Lower bounds and upper bounds obtained by GUROBI in a prespecified computation time are compared with the best feasible solutions found by the GRASP \times ILS heuristic. The numerical results show the promising performance of the proposed algorithm.

Further research will focus on designing a mechanism which only explores a part of potential bids but can still guarantee similar results as exploring the power-set of the set of tendered requests in order to reduce the computation time for bid generation.

Winner Determination Problem in Carrier Collaboration Via Combinatorial Exchange

Contents

| | | |
|------------|--|------------|
| 5.1 | Introduction | 88 |
| 5.2 | Literature review | 90 |
| 5.2.1 | Winner determination problems | 90 |
| 5.2.2 | Maximum vertex weight clique problems | 92 |
| 5.3 | Combinatorial exchange versus combinatorial auction | 93 |
| 5.3.1 | Better exploration of synergies among requests | 94 |
| 5.3.2 | More efficient allocation due to a larger solution space | 95 |
| 5.4 | Problem description and mathematical model | 97 |
| 5.4.1 | Formulation of the winner determination problem of the CE | 98 |
| 5.4.2 | Bidding languages of combinatorial exchange | 99 |
| 5.5 | Lagrangian relaxation approach | 100 |
| 5.5.1 | Framework of Lagrangian relaxation approach | 100 |
| 5.5.2 | Solving the Lagrangian relaxed problem | 104 |
| 5.5.3 | Update of Lagrangian multipliers | 109 |

| | | |
|------------|--|------------|
| 5.5.4 | Repair heuristic | 109 |
| 5.6 | Computational experiments | 111 |
| 5.6.1 | Generation of instances | 112 |
| 5.6.2 | Parameter setting | 115 |
| 5.6.3 | Experimental results | 115 |
| 5.7 | Conclusions | 118 |

5.1 Introduction

In previous two chapters, we have addressed the PDPTWPR (Chapter 3) and the stochastic BGP (Chapter 4) appeared in combinatorial auctions/exchanges. In this chapter, we will address the last key problem for transportation combinatorial auctions/exchanges appeared in the framework of this thesis (Section 1.2), the winner determination problem (WDP) in carrier collaboration via combinatorial exchange.

Generally, CTP must deal with two sub-problems, *i.e.*, reallocation of transport requests and profit sharing (Krajewska & Kopfer, 2006; Berger & Bierwirth, 2010; Dai & Chen, 2011; Robu, Noot, La Poutré, & Van Schijndel, 2011; Dai et al., 2014; Wang & Kopfer, 2014; Wang et al., 2014; Dai & Chen, 2015; Y. Li, Chen, & Prins, 2016). In this chapter, we focus on the first sub-problem : design of a combinatorial exchange (CE) mechanism to reallocate requests among carriers. This CE mechanism considers a scenario where each carrier tenders for acquiring (buying) requests from other carriers and selling requests to other carriers at the same time in order to maximize its profit. In the CE, carriers submit all their outsourcing requests to a virtual auctioneer and then requests are reallocated to them by the auc-

tioneer, based on all bids received. The bi-directional characteristic of the CE makes it able to exploit more potential profits by better exploration of synergies among the requests and more participation of the carriers. To the best of our knowledge, this topic was only conceptually addressed (Bloos & Kopfer, 2009; Ackermann et al., 2011) but never studied deeply in the literature.

The CE is an alternative transaction mechanism to combinatorial auctions (CAs) among less-than-truckload (LTL) carriers. In the CE, each carrier plays a double role of buyer and seller of transportation requests. This characteristic brings some advantages compared with traditional CA mechanisms in freight logistics. A 0-1 linear programming model is formulated for the problem. A Lagrangian relaxation approach is then developed to solve the winner determination problem of CE. The relaxed problem is transformed into a maximum vertex weight clique problem (MVWCP) which is solved either by a multi-neighborhood tabu search (MNTS) or by a commercial MIP solver. A repair heuristic is proposed to cope with any infeasibilities caused by the constraint relaxation. Extensive numerical experiments on randomly generated instances show that the Lagrangian relaxation approach can provide high quality solutions.

The contributions of this chapter are summarized as follows :

1. We introduce a new CE mechanism as an alternative method of CA mechanisms for requests reallocation/exchange among carriers in LTL transportation.
2. A formal mathematical model which can be adapted to distinct bidding languages is provided for the CE problem.
3. As the solution approach for the CE problem, an efficient approach

based on Lagrangian relaxation and on the MNTS algorithm for solving the clique problem is developed and proven to produce high quality solutions by numerical experiments on randomly generated instances.

This chapter is organized as follows. A brief literature review on the winner determination problem (WDP) and the MVWCP is given in Section 5.2. In Section 5.3, the advantages of the CE compared with CA mechanisms are discussed and illustrated by an easy-understood example. Section 5.4 provides a mathematical formulation of the CE. A Lagrangian relaxation approach is presented in Section 5.5. Computational experiments to evaluate the approach are reported in Section 5.6. Finally, Section 5.7 concludes this chapter with perspectives for future research.

5.2 Literature review

Our study is closely related to the winner determination problem (WDP) and the maximum vertex weight clique problem (MVWCP), so we will review the literature from the above two aspects.

5.2.1 Winner determination problems

The WDP in CA, also called combinatorial auction problem (CAP), is to determine the winning bidders and bids by the auctioneer. Since the problem is NP-hard (Rothkopf, Pekeč, & Harstad, 1998), both exact and heuristic methods have been developed to solve it.

Exact methods can solve the WDP to optimality but the computation time grows exponentially with problem size. The earliest attempt to exactly

solve the WDP can be found in Padberg (1973). Most exact algorithms apply branch-and-bound, such as the combinatorial auction structural search (CASS) (Fujishima, Leyton-Brown, & Shoham, 1999), the combinatorial auction multi-unit search (CAMUS) (Leyton-Brown, Shoham, & Tennenholtz, 2000), the BOB algorithm (Sandholm & Suri, 2003), the linear programming algorithm (Nisan, 2000) and the CABOB algorithm (Sandholm, Suri, Gilpin, & Levine, 2001). In addition, dynamic programming (Rothkopf et al., 1998), branch-and-price (Günlük, Ladányi, & De Vries, 2005) and branch-and-cut (Escudero, Landete, & Marín, 2009) techniques are also proposed to solve the WDP to optimality.

Some effective metaheuristics have also been developed since 2000, such as Casanova (Hoos & Boutilier, 2000), a simulated annealing approach (SAGII) (Hoos & Boutilier, 2000), a memetic algorithm (Boughaci, Benhamou, & Drias, 2009), tabu search algorithms (Boughaci, Benhamou, & Drias, 2010; Sghir, Hao, Jaafar, & Ghédira, 2014), and a multi-neighborhood tabu search (MNTS) algorithm (Wu & Hao, 2015b).

Lagrangian relaxation is also used to solve the WDP. Nandiraju et Regan (2004) apply a Lagrangian relaxation based algorithm to obtain near optimal solutions of the WDP in CA in the context of the procurement of transportation services. Guo, Lim, Rodrigues, et Tang (2006) convert a CA problem into an NP-hard set-packing problem and propose a Lagrangian heuristic algorithm to solve it. Kameshwaran, Benyoucef, et Xie (2005) present a progressive (multi-round) auction mechanism. Lagrangian relaxation is also used to solve a bid evaluation problem. Hsieh et Tsai (2008) and Hsieh (2010) consider a multi-round combinatorial reverse auction using the Lagrangian relaxation technique and the revelation of Lagrangian multipliers. Mansouri et Hassini (2015) implement a Lagrangian relaxation approach

to solve a problem of combinatorial multi-item multi-unit reverse auction (CMMRA).

5.2.2 Maximum vertex weight clique problems

Our CE problem is solved using a Lagrangian relaxation approach in which the relaxed problem is transformed into a maximum vertex weight clique problem (MVWCP). For this reason, we also review methods for solving the MVWCP and the related maximum clique problem (MCP). The MVWCP determines, in an undirected graph $G = (V, E)$ with vertex weights, a subset of pairwise adjacent nodes (*i.e.*, a clique) maximizing total weight.

The MCP, a particular case of the MVWCP with unit weights (Benlic & Hao, 2013), has a wide range of applications such as bioinformatics, cheminformatics, coding theory, economics, location, scheduling, social network analysis, and wireless networks (Wu & Hao, 2015a).

Like for the majority of combinatorial optimization problems, solution methods for the MVWCP can also be divided into exact methods and heuristic methods.

Babel (1994) introduces a branch-and-bound method calling a weighted coloring heuristic. Warren et Hicks (2006) present three distinct branch-and-bound methods based on the results of Balas et Yu (1986) and Babel (1994), where upper bounds and branching rules are derived from weighted clique covers.

As exact methods can only solve small instances of the MVWCP (Wu & Hao, 2015a), several heuristics are available to find near-optimal solutions in a reasonable amount of time. They include an augmentation algorithm

(Mannino & Stefanutti, 1999), a parallel and distributed heuristic based on replicator dynamics (Bomze, Pelillo, & Stix, 2000), a complementary pivoting algorithm (Massaro, Pelillo, & Bomze, 2002), a hybrid evolutionary approach (Singh & Gupta, 2006), a fast heuristic based on Motzkin-Straus theorem (Busygin, 2006), a phased local search algorithm (Pullan, 2008), a multi-neighborhood tabu search algorithm (Wu, Hao, & Glover, 2012) and a breakout local search (Benlic & Hao, 2013).

5.3 Combinatorial exchange versus combinatorial auction

Firstly, we briefly recall the set-packing formulation of the WDP in CA, presented in Ackermann et al. (2011).

$$\max \sum_{j \in B} p_j \cdot x_j \quad (5.1)$$

$$\sum_{j \in B} e_{jk} \cdot x_j \leq 1 \quad \forall k \in M \quad (5.2)$$

$$x_j \in \{0, 1\} \quad \forall j \in B \quad (5.3)$$

In this model, M denotes a set of requests and B a set of bids. Each bid j in B is associated with a price p_j . e_{jk} is a binary parameter equal to 1 if bid j includes request k and 0 otherwise. x_j is a binary variable indicating whether bid j is a winning bid.

The objective function (5.1) aims to maximize the auctioneer's revenue, *i.e.*, the total income of winning bids. Constraints (5.2) ensure that each request is assigned to at most one winning bid. In case that all requests must

be auctioned out, *i.e.*, there is no *free disposal*, each inequality in constraints (5.2) must be replaced by an equality (Andersson, Tenhunen, & Ygge, 2000; Sandholm, Suri, Gilpin, & Levine, 2002). Hereafter, model (5.1)-(5.3) is referred to as the CA model or problem.

Clearly, to implement a CA among carriers, each carrier must firstly determine and submit its outsourcing requests to an auctioneer. The carriers then bid for profitable requests in a common request pool held by the auctioneer. Bids are structured as bundles, *e.g.*, $\{(r_1, r_2, r_3), 50\}$ denotes a bid for acquiring (buying) requests r_1, r_2, r_3 of an offer price of 50.

Different from the bids in a traditional CA, each bid in CE consists of two parts, the buying-part and the selling-part, *e.g.*, $\{(+r_1, +r_2, -r_3, -r_4, -r_5), 100\}$ represents a bid for buying two requests r_1, r_2 with positive sign '+' and outsourcing (selling) three requests r_3, r_4, r_5 with negative sign '-', at a price of 100. All requests in one bid are linked, which means one bid can only be accepted or rejected as a whole in the WDP, and the price of a bid is not necessarily be positive. In the following, we explain why CE is more effective than CA for carrier collaboration in freight logistics.

5.3.1 Better exploration of synergies among requests

In a traditional CA, each carrier must determine and submit its outsourcing requests to a virtual auctioneer (*e.g.*, an auction platform) in the request offering phase and the carriers in coalition then bid for profitable requests in the bidding phase. In such a scenario, carriers can only bid for requests in the request pool held by the auctioneer. Whereas in CE, each carrier submits bids that contain both selling and buying requests, which

provides more possibilities for request exchange among bidders.

The primary reason to implement CE rather than CA is that the former provides more flexibility, as shown in the following example. Under a CA mechanism, assume that a carrier hosts request r_3 which is not profitable in its current routing plan. Meanwhile, two requests r_1, r_2 in the auctioneer's pool are profitable for this carrier. However, due to side constraints such as vehicle capacity and/or time windows, r_1, r_2, r_3 cannot be served simultaneously by the carrier.

In such a situation, one dilemma of static games with incomplete information (SGII) (Gibbons, 1992) occurs. On the one hand, bid $(+r_1, +r_2)$ can be infeasible if r_3 is not sold to another carrier once the WDP is solved : r_3 returns to its carrier who has not enough capacity to serve it with the winning requests $(+r_1, +r_2)$. On the other hand, missing the bid $(+r_1, +r_2)$ makes losing the opportunity if r_3 is acquired by another carrier. A CE mechanism can prevent from being trapped in such a dilemma by allowing the carrier to submit a bid $(+r_1, +r_2, -r_3)$ instead. In this case, the acquisition of r_1 and r_2 by winning the bid will be conditioned by the selling out of request r_3 .

5.3.2 More efficient allocation due to a larger solution space

An auction-based mechanism with a larger number of possible bids can lead to more efficient request allocations among carriers (Ackermann et al., 2011). It is interesting to note that CE can generate more bids than CA. For a carrier with p reserved requests (self-fulfillment requests) and a common pool with q requests, the number of possible bids that the carrier can submit is $2^q - 1$ in CA, versus $2^{p+q} - 1$ in CE. Of course, more potential bids may

require more computation time. One remedy addressed in [Buer \(2014\)](#) is to develop an effective way to reduce the number of bids, while guaranteeing the high-quality outcome of CE.

To further illustrate the advantages of the CE mechanism mentioned above, consider one simple instance of collaboration with two carriers X and Y and four requests. It is assumed that each carrier has a single vehicle, travel costs are equal to Euclidean distances, and request service times are negligible. Vehicle capacity is 7 units for X and 10 for Y . The other data are given in Table 5.1. Before collaboration, the transportation plan of each carrier is obtained by solving a vehicle routing problem with time windows (VRPTW), giving the two routes in Fig. 5.1 (a).

TABLE 5.1: Data for the illustrative example

| Node index | Description | Coordinates | Time window | Quantity | Price |
|------------|-------------------------------|-------------|-------------|----------|-------|
| X | Depot of carrier X | (0,10) | / | / | / |
| Y | Depot of carrier Y | (30,15) | / | / | / |
| 1 | Request 1 hosted by carrier Y | (10,0) | [10,50] | 1 | 50 |
| 2 | Request 2 hosted by carrier Y | (0,20) | [60,80] | 3 | 40 |
| 3 | Request 3 hosted by carrier X | (35,5) | [10,100] | 2 | 60 |
| 4 | Request 4 hosted by carrier X | (20,15) | [50,80] | 3 | 40 |

If a CA mechanism is adopted, carrier X can only bid for request 1 : $(+r_1)$. Indeed, if it bids for request 2, and requests 3, 4 are finally not served by carrier Y , carrier X has to serve three requests r_2, r_3 and r_4 , which violates the capacity constraint of its vehicle $(3 + 2 + 3 > 7)$. So, bids $(+r_2)$ and $(+r_1, +r_2)$ are not valid. The same analysis can be applied to carrier Y , who has only one feasible bid $(+r_3)$, since serving the three requests r_1, r_2 and r_4 by carrier Y via route $Y \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow Y$ would violate the time window constraint of r_4 (the same violation is inevitable for other routing plans in this situation).

On the other hand, if the CE mechanism is adopted, the carriers are

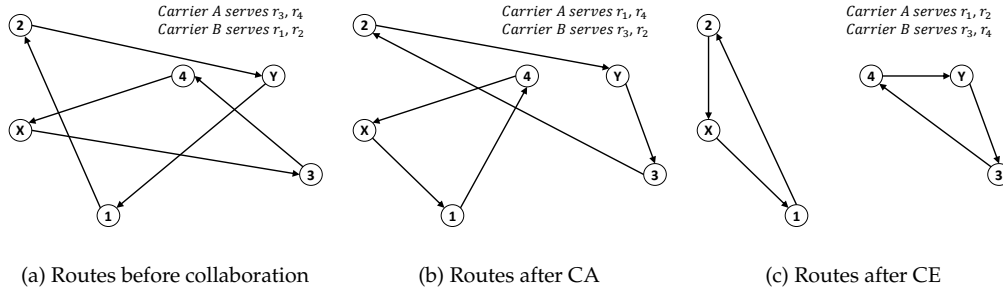


FIGURE 5.1: Comparison of routing plans among no-collaboration scenario, CA and CE.

TABLE 5.2: Comparison of results among no-collaboration scenario, CA and CE

| Situation | Cost | Revenue | Profit |
|---------------------------------|------|---------|--------|
| Carrier X without collaboration | 73.9 | 100 | 26.1 |
| Carrier Y without collaboration | 77.8 | 90 | 12.2 |
| Carrier X in CA | 52.8 | 90 | 37.2 |
| Carrier Y in CA | 79.7 | 100 | 20.3 |
| Carrier X in CE | 46.5 | 90 | 43.5 |
| Carrier Y in CE | 39.2 | 100 | 60.8 |

able to tender for more bids, which may generate more profits. For this example, the winning bids combination is $(+r_1, +r_2, -r_3, -r_4)$ for carrier X and $(+r_3, +r_4, -r_1, -r_2)$ for carrier Y.

Fig. 5.1 show the routes before collaboration (a), after CA (b) and after CE (c). Table 5.2 details the results of the three scenarios. CE outperforms clearly the other two schemes.

5.4 Problem description and mathematical model

Our CE can be viewed as a profit optimization scheme under the assumption that all carriers consent mutually to maximize their total profit as the primary task, whereas fair post-collaboration profit allocation among

the carriers is considered as a separate task. Actually, as long as the total profit of the carrier coalition increases, any carrier's individual profit will definitely not be lowered by the application of some reasonable profit sharing mechanism (Wang & Kopfer, 2014). The second issue is not addressed here since it goes beyond the scope of this research.

5.4.1 Formulation of the winner determination problem of the CE

Let N be a set of n carriers in a coalition, M the set of m requests they wish to exchange, and B a set of b bids. Each carrier i owns a set M_i of requests and submits a subset of bids B_i , hence $M = \bigcup_{i \in N} M_i$ and $B = \bigcup_{i \in N} B_i$ denotes the set of all bids. Each bid j has a price p_j , $\forall j \in B$. All bids submitted by one carrier i (B_i) are assumed to be feasible, *i.e.*, the requests that carrier i wants to sell are in M_i while the requests it wants to buy are in $M \setminus M_i$. To simplify the model, three sets of binary parameters are defined: e_{jk} equals 1 if and only if (iff) request k is in bid j , $a_{ik} = 1$ iff request k is in M_i (request k can be sold out by carrier i), and $b_{ik} = 1$ iff request k is in $M \setminus M_i$ (request k can be bought in by carrier i). The model involves also a binary variables x_j , equal to 1 iff j is a winning bid (*i.e.*, bid j is accepted by the auctioneer after solving the WDP).

$$\max \sum_{j \in B} p_j \cdot x_j \quad (5.4)$$

$$\sum_{i \in N} \sum_{j \in B_i} a_{ik} \cdot e_{jk} \cdot x_j \leq 1 \quad \forall k \in M \quad (5.5)$$

$$\sum_{i \in N} \sum_{j \in B_i} b_{ik} \cdot e_{jk} \cdot x_j \leq 1 \quad \forall k \in M \quad (5.6)$$

$$\sum_{i \in N} \sum_{j \in B_i} a_{ik} \cdot e_{jk} \cdot x_j = \sum_{i \in N} \sum_{j \in B_i} b_{ik} \cdot e_{jk} \cdot x_j \quad \forall k \in M \quad (5.7)$$

$$x_j \in \{0, 1\} \quad \forall j \in B \quad (5.8)$$

The objective function (5.4) represents the total price of all winning bids. Constraints (5.5) impose that each request appears at most once as a selling request in winning bids. In constraints (5.6), each request is included at most once as a buying request in winning bids. Constraints (5.7) are the sell-buy balance constraints, which state that any request is either not sold out or sold to only one winning bid. Hereafter, model (5.4)-(5.8) is referred to as the CE model or problem.

5.4.2 Bidding languages of combinatorial exchange

How to select a suitable bidding language for CA mechanisms in carrier collaboration is also a delicate problem. Generally, we have two choices for CA in the literature : the OR and XOR bidding languages (Nisan, 2000, 2006). XOR states that each carrier can win at most one bid while OR relaxes this constraint. Obviously, the tight constraint of XOR impacts negatively the quality of final solution and indirectly narrows bid construction space, whereas OR can often generate a more satisfactory outcome due to a broader solution space at the cost of supplementary computational burden. A compromising alternative is OR-of-XOR bidding language (Boutilier & Hoos, 2001), where carriers follow OR rule but each vehicle of carriers respects XOR regulation.

In this paper, we adopt OR bidding language for CE since OR is the

most general one among the three options. XOR and OR-of-XOR can also be easily implemented in the CE model since we can transform a relaxed problem of the model into the MVWCP, see Section 5.5.2.1.

5.5 Lagrangian relaxation approach

Lagrangian relaxation techniques transform a difficult optimization problem into a simpler one, by relaxing some hard constraints and moving them into the objective function. The relaxed constraints are penalized in case of violation by associating weights (Lagrangian multipliers) with them in the objective function. This dualization process yields a *Lagrangian relaxed problem* which is relatively easier to solve. The optimum of the *Lagrangian relaxed problem* for the given multipliers provides an upper bound on the optimum of the original maximization problem. The *Lagrangian dual problem* consists in determining the multipliers that minimize this bound.

5.5.1 Framework of Lagrangian relaxation approach

As in the CE model (Section 5.4), constraints (5.5) and (5.6) impose that each request must be sold out and bought in at most once, we name them *non-overlapping constraints*. The sell-buy balance constraints (5.7) are also called *matching constraints*.

The *matching constraints* are relaxed, which gives the WDP with a modified objective function. Let $\lambda = (\lambda_k)_{k \in M}$ be the *Lagrangian multipliers* associated with constraints (5.7). Then the *Lagrangian relaxed problem*, $RP(\lambda)$, can be formulated by equations (5.9)-(5.12). $Z(\lambda)$ is also referred to as the *Lagrangian dual function*. If $Z^*(\lambda)$ denotes its maximum for given multiplier

values, the *Lagrangian dual problem* can be formulated as $\min_{\lambda \in \mathbb{R}^m} Z^*(\lambda)$.

$$Z(\lambda) = \max \sum_{i \in N} \sum_{j \in B_i} p_j \cdot x_j + \sum_{k \in M} \lambda_k \cdot \left(\sum_{i \in N} \sum_{j \in B_i} a_{ik} \cdot e_{jk} \cdot x_j - \sum_{i \in N} \sum_{j \in B_i} b_{ik} \cdot e_{jk} \cdot x_j \right) \quad (5.9)$$

$$\sum_{i \in N} \sum_{j \in B_i} a_{ik} \cdot e_{jk} \cdot x_j \leq 1 \quad \forall k \in M \quad (5.10)$$

$$\sum_{i \in N} \sum_{j \in B_i} b_{ik} \cdot e_{jk} \cdot x_j \leq 1 \quad \forall k \in M \quad (5.11)$$

$$x_j \in \{0, 1\} \quad \forall j \in B \quad (5.12)$$

In $RP(\lambda)$, constraints (5.10) and (5.11) are similar to (5.2) in the CA model : both ensure that each request can only be traded at most once. So the relaxed problem looks like the WDP in CA. However, as the WDP is NP-hard and the Lagrangian relaxed problem must be solved many times, we decide to implement a fast tabu search metaheuristic explained in Section 5.5.2 instead of an exact method, to solve $RP(\lambda)$ without excessive running time even on large size instances. Indeed, [Zhao, Luh, et Wang \(1999\)](#) have proved that even if the relaxed problem is not optimally solved, the subgradient algorithm for solving the *Lagrangian dual problem* still converges to the optimal *Lagrangian multipliers* if the solution of the relaxed problem found at each iteration is close enough to its optimal solution.

The general structure of our Lagrangian relaxation approach is sketched in Algorithm 8 and its components are detailed in the next subsections. Roughly speaking, this is a subgradient procedure whose each iteration determines one upper bound, by solving the *Lagrangian relaxed problem* via a fast tabu search metaheuristic, and one feasible solution to the original CE

problem (giving a lower bound) by repairing the upper bounding solution. This procedure is followed by a post-optimization step where the *Lagrangian relaxed problem* with final values of multipliers is solved exactly, using a commercial MIP solver.

Starting with null multipliers, each main loop iteration (lines 2-18) begins by solving the *Lagrangian relaxed problem* with the current multipliers. To do so, $RP(\lambda)$ is converted into a maximum vertex weight clique problem (*MVWCP*, line 3), as explained in Section 5.5.2.1, which is then solved in line 4 using the multi-neighborhood tabu search (MNTS) described in Section 5.5.2.2. The metaheuristic returns a solution S_{MNTS} with total profit UB_{MNTS} . As the sequence of upper bounds is not always decreasing, a global best upper bound UB_{best} is updated in lines 5-7. If S_{MNTS} is feasible for the original CE problem (line 8), then the incumbent best feasible solution LB_{best} is updated by comparing with UB_{MNTS} . The next step in line 11 is to derive from S_{MNTS} a feasible solution S_{RH} to the original CE problem, via a repair heuristic RH (Section 5.5.4), giving a lower bound LB_{RH} . As the successive repairs do not necessarily yield increasing lower bounds, the best lower bound LB_{best} and the associated solution S_{best} are recorded in line 13. The relative gap between LB_{best} and UB_{best} is computed (line 15) and multipliers are adjusted (line 16) as shown in Section 5.5.3. The main loop stops after a maximum number of iterations $niter_{s_{LR}}$ or when the gap falls below a given threshold Gap . The post-optimization phase with the MIP solver is performed in lines 20-28.

Algorithm 8 – Overview of Lagrangian relaxation approach

```

1:  $iter \leftarrow 0, \lambda \leftarrow 0, UB_{best} \leftarrow \infty, LB_{best} \leftarrow -\infty, Gap \leftarrow +\infty$ 
2: repeat
3:   transform  $RP(\lambda)$  into a maximum vertex weight clique problem  $MVWCP$ 
4:   solve  $MVWCP$  using tabu search  $MNTS$ , giving solution  $S_{MNTS}$  with cost
    $UB_{MNTS}$  (Algorithm 9)
5:   if  $UB_{MNTS} < UB_{best}$  then
6:      $UB_{best} \leftarrow UB_{MNTS}$ 
7:   end if
8:   if  $S_{MNTS}$  is feasible for the original CE problem and  $LB_{best} < UB_{MNTS}$ 
   then
9:      $LB_{best} \leftarrow UB_{MNTS}, S_{best} \leftarrow S_{MNTS}$ 
10:  else
11:    repair  $S_{MNTS}$  using  $RH$  (Algorithm 11), giving a feasible solution  $S_{RH}$ 
    with cost  $LB_{RH}$ 
12:    if  $LB_{RH} > LB_{best}$  then
13:       $LB_{best} \leftarrow LB_{RH}, S_{best} \leftarrow S_{RH}$ 
14:    end if
15:     $Gap \leftarrow \frac{UB_{best} - LB_{best}}{UB_{best}}$ 
16:    update the vector of multipliers  $\lambda$  (Algorithm 10)
17:     $iter \leftarrow iter + 1$ 
18:  end if
19: until ( $iter = n_{iters_{LR}}$ ) or ( $Gap < 0.01$ )
20: solve  $RP(\lambda)$  via a MIP solver to get a solution  $S_{MIP}$  and its cost  $UB_{MIP}$ 
21: if  $UB_{MIP} < UB_{best}$  then
22:    $UB_{best} \leftarrow UB_{MIP}$ 
23: end if
24: repair  $S_{MIP}$  using  $RH$  to get a feasible solution  $S_{RH}$  and its cost  $L_{RH}$ 
25: if  $LB_{RH} > LB_{best}$  then
26:    $LB_{best} \leftarrow LB_{RH}, S_{best} \leftarrow S_{RH}$ 
27: end if
28:  $Gap \leftarrow \frac{UB_{best} - LB_{best}}{UB_{best}}$ 
29: return  $Gap, S_{best}, UB_{best}$  and  $LB_{best}$ 

```

5.5.2 Solving the Lagrangian relaxed problem

Ausiello, D'Atri, et Protasi (1980) explain how to reduce the set-packing problem to the MVWCP. Wu et Hao (2015b) apply this technique to transform the WDP in CA into a MVWCP, which is then solved using their MNTS heuristic (Wu et al., 2012). They hold so far the best results on 530 benchmark instances, both in terms running time and solution quality. Our *Lagrangian relaxed problem* differs from the WDP in CA by its two sets of packing constraints and its modified objective function, but we show in Section 5.5.2.1 that it is still possible to be converted into the MVWCP.

Initial tests unveiled excessive running time if $RP(\lambda)$ is solved exactly in each iteration of Algorithm 8. Using the MNTS brings a considerable speed-up (even if $RP(\lambda)$ must be converted into a MVWCP), while inducing a negligible upper bound increase on average. Moreover, repairing the upper bounding solutions obtained by the MNTS still provides good feasible solutions/lower bounds to the CE problem. However, to compensate the loss in solution quality and to obtain an exact upper bound without augmenting too much running time, we decide to solve $RP(\lambda)$ exactly, but only once at the end.

5.5.2.1 Transformation into the maximum vertex weight clique problem (MVWCP)

We show here how to transform our *Lagrangian relaxed problem* into the MVWCP. Consider a given CE problem with a set B of b bids. Each bid j in B is defined as a triplet (R_j^+, R_j^-, p_j) , where R_j^+ is the set of buying requests, R_j^- is the set of selling requests, and p_j denotes the price. An undirected graph $G = (V, E)$ can be constructed as follows :

- The node-set V contains one node j with weight p_j for each bid $j \in B$.
- The edge-set E contains one edge (j, k) if the two corresponding bids j and k are such that $R_j^+ \cap R_k^+ = \emptyset$ and $R_j^- \cap R_k^- = \emptyset$.

As two vertices (bids) connected by an edge have neither buying request nor selling request in common, a clique in G corresponds to a feasible solution of the *Lagrangian relaxed problem*, and a clique C with maximum total weight $W(C)$ corresponds to the optimal solution of the relaxed problem. Hence, any solution method for the MVWCP can be used to solve the *Lagrangian relaxed problem*. In general, the solution of the relaxed problem does not satisfy the *matching constraints* and is not feasible for the CE problem (5.4)-(5.8), but high quality solutions can be obtained by repairing the solutions of the relaxed problem in the follow-up repair procedure.

Consider one example of the CE problem with 2 carriers and 3 requests. Carrier 1 owns requests r_1 and r_3 , while carrier 2 has request r_2 , i.e., $N = \{1, 2\}$, $M = \{r_1, r_2, r_3\}$, $M_1 = \{r_1, r_3\}$, and $M_2 = \{r_2\}$. Table 5.3 gives the information of bids. The cliques associated with the optimum of $RP(0)$ (first *Lagrangian relaxed problem*) and the optimum of the CE problem respectively are depicted in Fig.5.2. $RP(0) = \{b_1, b_3, b_7, b_8\}$ has total profit 219 but is not feasible for the CE problem since no carrier sells request r_1 , which violates one *matching constraint*. The optimal solution of this instance is a combination of winning bids $\{b_1, b_3, b_6, b_8\}$ with total profit 210.

Since the edges between conflicting bids must not be included in the graph, other bidding languages like XOR and OR-of-XOR (mentioned in Section 5.4.2) can be directly implemented to our CE model, by removing the edges infeasible to them.

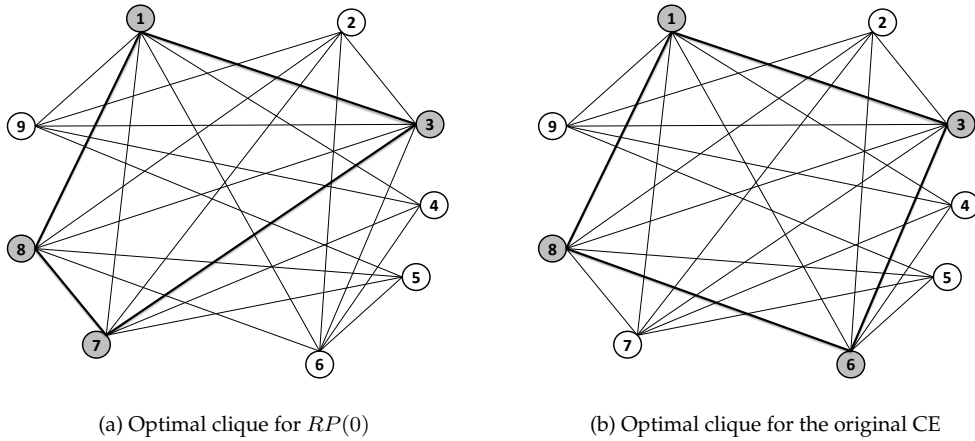


FIGURE 5.2: Comparison of optimal cliques between the relaxed problem and the CE problem.

TABLE 5.3: Bids information of the example

| Bid j | Bidder | Requests to sell R_j^- | Requests to buy R_j^+ | Price p_j |
|---------|-----------|--------------------------|-------------------------|-------------|
| 1 | Carrier 1 | $\{r_3\}$ | \emptyset | 65 |
| 2 | Carrier 1 | $\{r_1, r_3\}$ | \emptyset | 47 |
| 3 | Carrier 1 | \emptyset | $\{r_2\}$ | 63 |
| 4 | Carrier 1 | $\{r_1\}$ | $\{r_2\}$ | -18 |
| 5 | Carrier 1 | $\{r_1, r_3\}$ | $\{r_2\}$ | 79 |
| 6 | Carrier 2 | $\{r_2\}$ | \emptyset | 13 |
| 7 | Carrier 2 | $\{r_2\}$ | $\{r_1\}$ | 22 |
| 8 | Carrier 2 | \emptyset | $\{r_3\}$ | 69 |
| 9 | Carrier 2 | $\{r_2\}$ | $\{r_1, r_3\}$ | -10 |

5.5.2.2 Multi-neighborhood tabu search (MNTS)

To solve the MVWCP, we directly implement the MNTS algorithm proposed by Wu et al. (2012), briefly presented as in Algorithm 9. The reader is referred to Wu et al. (2012) for more details.

Before each call of the *MNTS* to solve the Lagrangian relaxed problem, $RP(\lambda)$ is reduced to a MVWCP defined on an undirected weighted graph $G = (V, E)$ in Algorithm 8. The algorithm returns a clique $C_{GlobalBest}$ with

maximum or nearly maximum total weight $W(C_{GlobalBest})$.

Initial solution construction :

The initial solution is constructed using an iterative procedure. One seed-vertex is first randomly selected to generate an initial clique C . Then, at each iteration, a new vertex j is randomly selected among all non-conflicting vertices and inserted into C , i.e., $j \notin C$, and j is connected to all other vertices of C . The procedure stops when no more vertex can be added. This randomized heuristic is fast, easy to implement and generates diversified solutions.

Neighborhoods used :

Three neighborhoods called N_{swap} , N_{add} and N_{drop} are browsed to jointly improve the current clique C . In each tabu search iteration, the best non-tabu neighbor is determined (even if the total weight of the corresponding clique decreases) to replace the incumbent solution.

The moves defining N_{swap} exchange one vertex i in C with one vertex j in $V \setminus C$, such that $(i, j) \notin E$ and j is connected to all but one vertices in C , i.e., $|A(j) \cap C| = |C| - 1$, where $A(j)$ is the set of adjacent vertices of j in G . N_{add} is defined by the moves that add one vertex i to clique C , i.e., $i \in V \setminus C$ and $(i, v) \in E, \forall v \in C$. Finally, N_{drop} simply removes one vertex i from clique C . This last move is useful since the weight of a vertex (the price of a bid) may be negative in our problem.

Tabu list and tabu tenure :

A prohibition rule is used in Algorithm 9 : Once a vertex leaves the current clique C , it is banned to come back during the next $|C| + LengthTabu$ iterations, for the neighborhood N_{swap} , and during $LengthTabu$ iterations, for the neighborhood N_{drop} , where $LengthTabu$ is a tabu tenure parameter.

Multi-start mechanism and stopping criteria :

A restart is triggered after a given number of iterations without improvement. The number is denoted as $DepthSearch$. Each restart calls the randomized heuristic to construct a new initial solution. Algorithm 8 stops when the total number of iterations over successive restarts reaches a given maximum number $niter_{MNTS}$.

Algorithm 9 – Multi-neighborhood tabu search MNTS

```

1:  $iter \leftarrow 0, C_{GlobalBest} \leftarrow \emptyset$ 
2: while  $iter < niter_{MNTS}$  do
3:   initialize current clique  $C$  using the randomized constructive heuristic
4:   reset tabu list
5:    $notbetter \leftarrow 0$ 
6:    $C_{LocalBest} \leftarrow C$ 
7:   while ( $notbetter < DepthSearch$ ) and ( $iter < niter_{MNTS}$ ) do
8:      $iter \leftarrow iter + 1$ 
9:     find the best non-tabu clique  $C'$  in  $N_{swap} \cup N_{add} \cup N_{drop}$ 
10:     $C \leftarrow C'$ 
11:    update tabu list
12:     $iter_{Restart} \leftarrow iter_{Restart} + 1$ 
13:    if  $W(C) > W(C_{LocalBest})$  then
14:       $C_{LocalBest} \leftarrow C$ 
15:       $notbetter \leftarrow 0$ 
16:    else
17:       $notbetter \leftarrow notbetter + 1$ 
18:    end if
19:  end while
20:  if  $W(C_{LocalBest}) > W(C_{GlobalBest})$  then
21:     $C_{GlobalBest} \leftarrow C_{LocalBest}$ 
22:  end if
23: end while
24: return  $C_{GlobalBest}$ 

```

Although *the matching constraints* are relaxed, they are sometimes satisfied during the execution of the MNTS, giving a feasible solution of the CE problem. In such a case, if the *Lagrangian relaxed problem* $RP(\lambda)$ is solved to

optimality, we would have an optimal solution of the original problem and could stop the MNTS. However, the MNTS is a heuristic algorithm, and does not necessarily solve $RP(\lambda)$ to optimality. That's why the MNTS must continue until reaching its maximum number of iterations.

5.5.3 Update of Lagrangian multipliers

Algorithm 10 shows how the Lagrangian multipliers $\lambda = (\lambda_k)_{k \in M}$ are updated in line 16 of Algorithm 8. The notation δ denotes a given step size, $\delta \leq 2$. Recall that m is the number of requests.

Algorithm 10 – Update of Lagrangian multipliers

- 1: $t \leftarrow \frac{\delta \cdot (UB_{MNTS} - LB_{RH})}{\|\sum_{i \in N} \sum_{j \in B_i} \sum_{k \in M} x_j \cdot e_{jk} \cdot (a_{ik} - b_{ik})\|^2}$
 - 2: **for** $k \leftarrow 1$ **to** m **do**
 - 3: $\lambda_k \leftarrow \lambda_k + t \cdot \left(\sum_{i \in N} \sum_{j \in B_i} \sum_{k \in M} x_j \cdot e_{jk} \cdot (a_{ik} - b_{ik}) \right)$
 - 4: **end for**
-

5.5.4 Repair heuristic

Algorithm 11 sketches the repair heuristic used to derive a feasible solution to the original CE problem (S_{RH}) from the solution obtained by the MNTS (S_{MNTS}).

A solution of the *Lagrangian relaxed problem* is infeasible to the original CE problem because some requests in the solution do not meet the *matching constraints*. Such a request is called *no-balanced* iff it is included in a winning bid of the relaxed problem while its corresponding *matching constraint* is violated. So, the idea to repair an infeasible solution is to bring more bids to the final solution which makes all *no-balanced* requests meet their *matching constraints*.

More precisely, if a request k in the solution of the *Lagrangian relaxed problem* makes $\sum_{i \in N} \sum_{j \in B_i} a_{ik} \cdot e_{jk} \cdot x_j - \sum_{i \in N} \sum_{j \in B_i} b_{ik} \cdot e_{jk} \cdot x_j > 0$ (Constraint (5.7)), then request k is included in the set R_{nopair}^- . Conversely, if a request k makes $\sum_{i \in N} \sum_{j \in B_i} a_{ik} \cdot e_{jk} \cdot x_j - \sum_{i \in N} \sum_{j \in B_i} b_{ik} \cdot e_{jk} \cdot x_j < 0$, then request k is included in the other set R_{nopair}^+ . $R_{nopair} = \{R_{nopair}^+, R_{nopair}^-\}$ consists of two sets of *no-balanced* requests in the solution of the *Lagrangian relaxed problem* that violate the *matching constraints* of the original CE problem. For example, assume that $\{b_1, b_2\}$ is a winning bids combination of the *Lagrangian relaxed problem*, where $b_1 = \{(+r_1, -r_2), p_{b_1}\}$ and $b_2 = \{(+r_3, -r_1), p_{b_2}\}$. Then we have $R_{nopair}^+ = \{+r_3\}$, $R_{nopair}^- = \{-r_2\}$, and $R_{nopair} = \{\{+r_3\}, \{-r_2\}\}$.

Let $R_{matching} = \{R_{matching}^+, R_{matching}^-\}$ be two complementary sets of R_{nopair} , where $R_{nopair}^+ = R_{matching}^-$, $R_{nopair}^- = R_{matching}^+$. For the above example, $R_{matching} = \{\{+r_2\}, \{-r_3\}\}$.

To simplify the presentation of the RH algorithm, we give the following two additional definitions : i) A bid $b \in R_{matching}$ iff $R_b^+ \subseteq R_{matching}^+$ and $R_b^- \subseteq R_{matching}^-$, where R_b^+ , R_b^- are the set of buying requests and the set of selling requests of bid b , respectively. ii) A bid set $B = R_{matching}$ iff $\cup_{b \in B} R_b^+ = R_{matching}^+$ and $\cup_{b \in B} R_b^- = R_{matching}^-$.

The repair algorithm returns a feasible solution S_{RH} with its cost LB_{RH} of the original CE problem.

Note that the repair algorithm returns in most cases a feasible solution S_{RH} of the original CE problem and its cost LB_{RH} . Sometimes it can fail but this is not a problem : in that case Algorithm 11 indicates the failure by setting LB_{RH} to $-\infty$ and the best lower bound LB_{best} is not updated in line 13 of Algorithm 8. On the other hand, although the number of subsets tested in line 8 grows exponentially with instance size, it still remains small

even for large size instances (Section 5.6.1) and so the repair heuristic is very fast in practice.

Algorithm 11 – Repair heuristic – $RH(S)$

```

1: generate  $R_{nopair}$  from the infeasible solution  $S$ 
2: generate  $R_{matching}$  from  $R_{nopair}$ 
3: for all bids  $b_{repair} \in B$  do
4:   if  $b_{repair} \notin S$  and  $b_{repair} \in R_{matching}$  then
5:     include bid  $b_{repair}$  in the set of bids  $B_{repair}$ 
6:   end if
7: end for
8: for all possible subsets  $\underline{B}_{repair} \subseteq B_{repair}$  do
9:   if  $\underline{B}_{repair} = R_{matching}$  then
10:    record  $\underline{B}_{repair}$  in a set  $L$ 
11:   end if
12: end for
13: sort all elements in set  $L$  in non-increasing order of their total profit
14: denote the first element in the sorted set  $L$  as  $B_{repair}^{best}$ 
15:  $S_{RH} \leftarrow S \cup B_{repair}^{best}$ 
16: return  $S_{RH}$  with cost  $LB_{RH}$ 

```

5.6 Computational experiments

All experiments were conducted on a desktop equipped with an Intel Core i7-2600 3.40 GHz processor, 8 GB RAM, and Windows 7 Professional (64 bits) service pack 1. The Lagrangian relaxation algorithm and the instance generator described in the sequel were both coded in Python programming language (Interpreter : Cpython 3.4.3, IDE : JetBrains PyCharm 4.5.4). The 0-1 linear programming model for the CE problem was built and solved using the GUROBI MIP solver (version 6.5.0) and its API for Python, using default parameters.

5.6.1 Generation of instances

Since the CE problem was rarely studied in the literature, we could not find existing instances and decided to develop an instance generator by ourselves, to evaluate the performance of the Lagrangian relaxation approach but also for future research of the CE problem. This instance generator in Python can be found via : <https://github.com/yuan296103/CE-instances-generator>, and the LTL requests data used in the simulation tests are extracted from the benchmark instances in [Ropke et Cordeau \(2009\)](http://www.diku.dk/~sropke/), available at <http://www.diku.dk/~sropke/>.

The user can generate *complete-bids* (C) or *partial-bids* (P) instances. In both cases, the number of carriers m and the number of requests n are given. Then each request is assigned to carriers one by one. For request k , it is assigned to carrier i with a probability of $P_{ik} = \frac{1/D_{is_{ik}}}{\sum_{i \in N} 1/D_{is_{ik}}}$, where $D_{is_{ik}}$ represents the transportation cost/traveled distance for carrier i to serve request k individually. Thus, for each carrier, its own request set M_i is defined. We use this method to initially allocate requests among carriers since the manner well simulates the realistic situation, *i.e.*, the closer a carrier is to a request, the more likely for the carrier to get the request.

In complete-bids instances, each carrier can bid for any requests from other carriers and submit all possible bids. Hence, the number of possible bids submitted by carrier i is $2^{|M_i|} \cdot 2^{n-|M_i|} - 1$ and the total number of bids b is $m \cdot (2^n - 1)$. In partial-bids instances, the number of bids b is fixed and each carrier can only submit $\lfloor b/m \rfloor$ bids, randomly selected from all possible bids.

In the request-based bid price generation, we use two options to generate the price for each request. One takes into account both distance fac-

tor and quantity factor (denoted by option A), the other one considers distance factor solely (denoted by option B). For the first option, each request i is set with a selling price $p_{sell}^i = \rho_{sell} \times distance^i \times load^i$ and a buying price $p_{buy}^i = \rho_{buy} \times distance^i \times load^i$, whereas for the second option each request i is set with a selling price $p_{sell}^i = \rho_{sell} \times distance^i$ and a buying price $p_{buy}^i = \rho_{buy} \times distance^i$. $distance^i$ represents the distance between the pickup node and the delivery node of request i and $load^i$ represents the demand quantity of each request i . The factor ρ_{sell} and ρ_{buy} are randomly generated according to the uniform distribution defined on $(0, 0.5)$ and $(0, 1)$. p_{sell}^i can be interpreted as the price that a carrier will charge if it outsources (sells) request i to other carriers, and p_{buy}^i can be interpreted as the price that a carrier will pay if it acquires (buys) request i from other carriers and serves the request. p_{buy}^i has a wider value interval than p_{sell}^i , because we want to make the prices of most bids in an instance be positive, which more conforms to realistic scenarios.

Then the price of each bid j can be set as $p_j = \sum_{i \in R_j^+} p_{buy}^i \cdot \sigma - \sum_{i \in R_j^-} p_{sell}^i$, where σ is a synergy factor randomly generated from the interval $[1, 1 + (m_j^+ - 1)/2m_j^+]$, with $m_j^+ = |R_j^+|$ being the number of requests to buy in bid j . Why the synergy factor is generated in this way is explained as follows. Assume that all requests in R_j^+ are delivery requests (delivery of goods from a depot to m_j^+ customers) and the distance from each customer to the depot and the distance between any two customers are both the unit distance on the Euclidean plane. In case that each request in R_j^+ is served individually, the total cost (distance) for serving the m_j^+ requests is $2m_j^+$. In case that all requests in R_j^+ are served by the same carrier on the same route, the total cost (distance) for serving the requests is $m_j^+ + 1$. Compared with the first extreme case, the synergy among the requests in the second

extreme case can reduce their total service cost by $[(m_j^+ - 1)/2m_j^+] \cdot 100\%$, where $(m_j^+ - 1)/2m_j^+ = 1 - (m_j^+ + 1)/2m_j^+$. This cost reduction will be turned into profit increase. We assume that the percentage of profit increase is the same as (or close to) the percentage of cost reduction. Then in the second extreme case, the total profit for serving all requests in R_j^+ will be $\sum_{i \in R_j^+} p_{buy}^i \cdot [1 + (m_j^+ - 1)/2m_j^+]$. Obviously, the total profit for serving the requests by the same carrier will never be lower than the total profit for serving each of the requests individually, *i.e.*, $\sum_{i \in R_j^+} p_{buy}^i$. So we can generate the synergy factor σ from the interval $[1, 1 + (m_j^+ - 1)/2m_j^+]$. Of course, this way of generating σ is heuristic, but it is intuitively reasonable, because $\sigma = 1$ when $m_j^+ = 1$ (in this case, $1 + (m_j^+ - 1)/2m_j^+ = 1$), $1 + (m_j^+ - 1)/2m_j^+$ increases when m_j^+ increases, and $1 + (m_j^+ - 1)/2m_j^+ \rightarrow 1.5$ when $m_j^+ \rightarrow +\infty$. Note that the larger the factor σ , the higher the synergy among the requests.

Each instance is named with the format of $n - m - b - C/P - A/B - original\ instance$, where n is the number of carriers, m is the number of requests, b is the number of bids, C/P indicates complete/partial bids, A/B represents the method to generate requests' prices, and *original instance* indicates the instance name used to generate requests' prices. For example, 2-4-30-C-B-AA30 is an instance with 2 carriers, 4 requests, 30 bids, complete bids, requests' prices generated by option B, and using AA30 to generate requests' prices. We generated three sets of instances : 10 small instances (up to 1000 bids), 20 medium instances (1001-5000 bids) and 20 large instances (5001-12000 bids).

5.6.2 Parameter setting

The Lagrangian relaxation algorithm is controlled by relatively few parameters. Their values were selected in preliminary experiments to achieve a good trade-off between solution quality and CPU time. As shown in Table 5.4, they depend on instance size.

TABLE 5.4: Parameter setting according to instance size

| Symbol | Role | Small | Medium | Large |
|-----------------|--|-------|--------|-------|
| $niters_{LR}$ | Maximum number of iterations of LR | 10 | 100 | 200 |
| $niters_{MNTS}$ | Maximum number of iterations of $MNTS$ | 1000 | 10000 | 20000 |
| δ | Step size to update Lagrangian multipliers | 1 | 1.5 | 1.9 |
| $LengthTabu$ | Length of tabu tenure | 7 | 8 | 9 |
| $DepthSearch$ | Parameter for triggering the restart of $MNTS$ | 20 | 30 | 50 |

5.6.3 Experimental results

In this section, we present the computational results and compare the performance of our Lagrangian relaxation approach with that of the MIP solver. One table of results is provided for each instance set (small, medium and large). For the Lagrangian approach are reported the upper bound UB , the lower bound LB , the relative duality gap $\frac{UB-LB}{UB}$ and the running time T in seconds. For the solver, the tables display the upper bound (only when no optimum is found), the lower bound (optimal or best feasible solution found), and the running time, with a time limit of 3600 seconds. In addition, the saving by percentage achieved by the Lagrangian relaxation approach over the MIP solver is given as $Imp = \frac{LB(LR)-LB(MIP)}{LB(MIP)}$. An asterisk in column 7 and a bold face font in column 3 indicate that the solutions are proven to be optimal by the MIP solver.

Table 5.5 reports the results on the 10 small size instances. The solver finds in all cases an optimal solution, in less than one second (0.28 seconds

on average). The Lagrangian heuristic is only a bit slower (0.39 seconds on average). Its lower and upper bounds are never equal but the difference never exceeds 1% (0.47% on average) and, in fact, all the optima found by the solver are retrieved. These results show that both solution methods are very efficient on small instances.

TABLE 5.5: Lagrangian relaxation versus MIP on small size instances

| Instance | Lagrangian relaxation | | | | MIP | | | | Imp |
|--------------------|-----------------------|----------------|------------|----------|-----------|-----------|------------|----------|-----|
| | <i>UB</i> | <i>LB</i> | <i>Gap</i> | <i>T</i> | <i>UB</i> | <i>LB</i> | <i>Gap</i> | <i>T</i> | |
| 2-4-30-C-A-AA30 | 643.91 | 640.86 | 0.47 | 0.02 | - | *640.86 | - | 0.04 | 0 |
| 2-4-30-C-B-BB30 | 87.65 | 87.33 | 0.37 | 0.01 | - | *87.33 | - | 0.04 | 0 |
| 3-6-189-C-A-AA30 | 726.74 | 721.64 | 0.70 | 0.15 | - | *721.64 | - | 0.33 | 0 |
| 3-6-189-C-B-CC30 | 103.47 | 102.76 | 0.69 | 0.23 | - | *102.76 | - | 0.05 | 0 |
| 3-8-300-P-A-BB30 | 862.27 | 861.94 | 0.04 | 0.31 | - | *861.94 | - | 0.62 | 0 |
| 3-8-300-P-B-AA30 | 119.63 | 119.46 | 0.14 | 0.45 | - | *119.46 | - | 0.21 | 0 |
| 4-7-508-C-A-AA50 | 527.33 | 526.53 | 0.15 | 0.33 | - | *526.53 | - | 0.22 | 0 |
| 4-7-508-C-B-AA50 | 133.07 | 131.18 | 1.42 | 0.37 | - | *131.18 | - | 0.22 | 0 |
| 4-10-1000-P-A-XX30 | 1514.68 | 1510.94 | 0.25 | 1.04 | - | *1510.94 | - | 0.54 | 0 |
| 4-10-1000-P-B-YY30 | 252.66 | 251.43 | 0.49 | 0.99 | - | *251.43 | - | 0.53 | 0 |
| Average | - | - | 0.47 | 0.39 | - | - | - | 0.28 | 0 |

Table 5.6 summarizes the experimental results for the 20 medium size instances. GUROBI is optimal on 10 instances but not on the other 10 partial-bids instances (code P) with at least 40 requests and 4000 bids, even in one hour of computation. Its average gap and running time are respectively 2.25% and 2020.37 seconds. Our Lagrangian relaxation algorithm finds not only the 10 proven optima but obtains better solutions than the MIP solver in the other cases, with a 1.66% higher profit on average (for the instances which are not solved to optimality by the MIP solver). It looks more stable both in terms of gap (average 0.50%, maximum 1.59%) and speed (average 75.82 seconds). Summarizing, on average, the Lagrangian approach runs faster and finds better solutions than the solver on medium instances.

The results of the 20 large size instances, are given in Table 5.7. The sol-

TABLE 5.6: Lagrangian relaxation versus MIP on medium size instances

| Instance | Lagrangian relaxation | | | | MIP | | | | Imp |
|---------------------|-----------------------|----------------|------------|----------|-----------|-----------|------------|----------|------|
| | <i>UB</i> | <i>LB</i> | <i>Gap</i> | <i>T</i> | <i>UB</i> | <i>LB</i> | <i>Gap</i> | <i>T</i> | |
| 4-8-1020-C-A-BB30 | 3462.17 | 3460.59 | 0.05 | 1.92 | - | *3460.59 | - | 0.52 | 0 |
| 4-8-1020-C-B-XX35 | 139.57 | 138.64 | 0.67 | 2.02 | - | *138.64 | - | 0.41 | 0 |
| 2-20-2000-P-A-BB35 | 5723.61 | 5717.43 | 0.26 | 3.14 | - | *5717.43 | - | 12.44 | 0 |
| 2-20-2000-P-B-CC35 | 480.6 | 479.34 | 0.44 | 3.60 | - | *479.34 | - | 20.49 | 0 |
| 5-9-2555-C-A-AA55 | 1260.42 | 1251.89 | 0.68 | 7.48 | - | *1251.89 | - | 5.47 | 0 |
| 5-9-2555-C-B-AA55 | 116.31 | 115.06 | 1.07 | 7.27 | - | *115.06 | - | 5.29 | 0 |
| 3-30-3000-P-A-DD30 | 8390.24 | 8385.12 | 0.06 | 32.55 | - | *8385.12 | - | 1391.83 | 0 |
| 3-30-3000-P-B-CC50 | 706.54 | 702.81 | 0.53 | 30.49 | - | *702.81 | - | 2966.36 | 0 |
| 4-40-4000-P-A-AA55 | 8491.38 | 8477.41 | 0.16 | 50.63 | 8582.94 | 8465.93 | 1.36 | 3600 | 0.14 |
| 4-40-4000-P-B-BB55 | 776.81 | 772.30 | 0.58 | 42.84 | 774.05 | 765.62 | 1.09 | 3600 | 0.87 |
| 5-40-4000-P-A-AA65 | 8139.92 | 8109.12 | 0.38 | 41.04 | 8144.16 | 7982.56 | 1.98 | 3600 | 1.59 |
| 5-40-4000-P-B-BB65 | 1015.18 | 1012.98 | 0.22 | 46.93 | 1029.61 | 999.94 | 2.88 | 3600 | 1.30 |
| 4-10-4092-C-A-XX40 | 143.35 | 142.58 | 0.54 | 41.17 | - | *142.58 | - | 2.26 | 0 |
| 4-10-4092-C-B-YY40 | 266.74 | 263.68 | 1.15 | 52.19 | - | *263.68 | - | 2.28 | 0 |
| 8-45-4600-P-A-XX45 | 888.77 | 874.68 | 1.59 | 153.64 | 881.77 | 851.29 | 3.46 | 3600 | 2.75 |
| 8-45-4600-P-B-YY45 | 981.06 | 976.68 | 0.45 | 165.70 | 982.79 | 949.19 | 3.42 | 3600 | 2.90 |
| 5-50-5000-P-A-AA60 | 10454.80 | 10413.51 | 0.39 | 187.43 | 10414.87 | 10318.91 | 0.92 | 3600 | 0.92 |
| 5-50-5000-P-B-BB60 | 1015.3 | 1011.69 | 0.36 | 201.21 | 1013.21 | 1005.51 | 0.76 | 3600 | 0.61 |
| 10-50-5000-P-A-CC60 | 10237.08 | 10226.92 | 0.10 | 227.73 | 10265.17 | 10160.27 | 1.02 | 3600 | 0.66 |
| 10-50-5000-P-B-DD60 | 1116.44 | 1113.48 | 0.27 | 217.44 | 1124.92 | 1061.63 | 5.63 | 3600 | 4.88 |
| Average | - | - | 0.50 | 75.82 | - | - | 2.25 | 2020.37 | 1.66 |

ver is still able to solve 4 instances to optimality with a relative low number of requests, although their number of bids exceeds 5000. Nevertheless, it fails to solve the remaining 16 partial-bids instances with a large number of requests : even after one hour of execution, the solver gap is 5.21% on average and even reaches 11.75%. Once again, the Lagrangian approach retrieves the proven optima and in the other cases improves GUROBI results by 3.91% on average. Its gaps increase compared with medium instances (average 1.14%, maximum 2.76%), but they are still relatively small and obtained in reasonable computation time (829.95 seconds on average).

A few remarks result from the above simulation tests. GUROBI is more efficient than the Lagrangian relaxation algorithm for the CE instances with a low number of requests, but our Lagrangian algorithm significantly outperforms the solver both in terms of computation time and solution quality for the instances with a large number of requests (more than 40). In fact, some previous studies pointed out that commercial MIP solvers are often

TABLE 5.7: Lagrangian relaxation versus MIP on large size instances

| Instance | Lagrangian relaxation | | | | MIP | | | | Imp |
|----------------------|-----------------------|----------------|------------|----------|-----------|-----------|------------|----------|-------|
| | <i>UB</i> | <i>LB</i> | <i>Gap</i> | <i>T</i> | <i>UB</i> | <i>LB</i> | <i>Gap</i> | <i>T</i> | |
| 5-10-5115-C-A-AA70 | 1716.53 | 1699.85 | 0.97 | 153.22 | - | *1699.85 | - | 3.83 | 0 |
| 5-10-5115-C-B-BB70 | 284.03 | 280.87 | 1.11 | 166.80 | - | *280.87 | - | 3.19 | 0 |
| 6-50-6000-P-A-XX70 | 1114.49 | 1099.84 | 1.32 | 285.53 | 1121.04 | 989.30 | 11.75 | 3600 | 10.05 |
| 6-50-6000-P-B-XX75 | 1094.07 | 1087.53 | 0.60 | 226.57 | 1088.71 | 1061.69 | 2.48 | 3600 | 2.43 |
| 4-40-6000-P-A-YY70 | 758.30 | 753.93 | 0.58 | 316.32 | 728.75 | 723.24 | 0.76 | 3600 | 4.07 |
| 4-40-6000-P-B-YY75 | 889.15 | 879.37 | 1.10 | 246.18 | 888.18 | 868.52 | 2.21 | 3600 | 1.25 |
| 5-50-7000-P-A-AA50 | 10536.97 | 10494.32 | 0.40 | 411.63 | 10556.10 | 10418.72 | 1.30 | 3600 | 0.72 |
| 5-50-7000-P-B-BB50 | 808.97 | 807.91 | 0.13 | 435.76 | 809.55 | 800.22 | 1.15 | 3600 | 0.96 |
| 4-11-8118-C-A-CC70 | 3066.30 | 3045.06 | 0.69 | 607.24 | - | *3045.06 | - | 4.84 | 0 |
| 4-11-8118-C-B-DD70 | 235.11 | 234.11 | 0.43 | 658.19 | - | *234.11 | - | 4.82 | 0 |
| 7-50-8400-P-A-CC50 | 10835.73 | 10550.77 | 2.63 | 717.39 | 10676.16 | 10101.59 | 5.38 | 3600 | 4.26 |
| 7-50-8400-P-B-DD50 | 849.01 | 845.04 | 0.47 | 818.67 | 884.53 | 825.39 | 6.69 | 3600 | 2.38 |
| 8-55-9600-P-A-AA60 | 9710.22 | 9632.60 | 0.80 | 1006.12 | 9704.21 | 9115.74 | 6.06 | 3600 | 5.37 |
| 8-55-9600-P-B-BB60 | 1455.96 | 1426.46 | 2.03 | 1210.81 | 1427.33 | 1410.87 | 1.15 | 3600 | 1.10 |
| 10-60-10000-P-A-CC60 | 13003.43 | 12874.78 | 0.99 | 1453.84 | 13016.09 | 11786.76 | 9.44 | 3600 | 8.45 |
| 10-60-10000-P-B-DD60 | 1357.12 | 1346.31 | 0.80 | 1226.57 | 1376.19 | 1329.33 | 3.41 | 3600 | 1.28 |
| 12-65-12000-P-A-AA65 | 13217.40 | 12852.32 | 2.76 | 1389.42 | 13178.75 | 12237.53 | 7.14 | 3600 | 4.78 |
| 12-65-12000-P-B-BB65 | 1536.93 | 1515.89 | 1.37 | 1658.40 | 1595.3 | 1475.17 | 7.53 | 3600 | 2.76 |
| 12-65-12000-P-A-CC70 | 12764.73 | 12619.95 | 1.13 | 2004.17 | 12690.57 | 11425.88 | 9.97 | 3600 | 9.46 |
| 12-65-12000-P-B-DD70 | 1358.72 | 1324.32 | 2.53 | 1606.24 | 1377.68 | 1282.72 | 6.89 | 3600 | 3.24 |
| Average | - | - | 1.14 | 829.95 | - | - | 5.21 | 2880.81 | 3.91 |

able to solve the WDP in CA, but the computation time largely depends on the request and bid structure. For example, the CA problems in the well-known set CATS (Leyton-Brown, Pearson, & Shoham, 2000) have a huge number of bids for each instance but a small number of items (requests in our CE) per bid : MIP solvers obtain optimal solutions in less than one second. The instances of another set REL (Lau & Goh, 2002) have roughly the same number of items and number of bids for each instance, and in that case commercial solvers are dominated by dedicated algorithms. These results from previous experiments concerning the WDP in CA might explain our conclusion.

5.7 Conclusions

Requests reallocation/exchange is a key issue for carrier collaboration in freight logistics. In this chapter, we propose an alternative mechanism

which differs from traditional combinatorial auctions (CAs) for the exchange of transportation requests among carriers. The new combinatorial exchange (CE) mechanism shows some advantages compared with CA : a better exploration of possible synergies among requests and a sufficient participation from carriers.

We provided a 0-1 linear programming model for the CE problem and design a Lagrangian relaxation approach to solve it. The hard *matching constraints* are relaxed but the *Lagrangian relaxed problem* is still equivalent to a NP-hard maximum vertex weight clique problem (MVWCP). However, optimal or quasi-optimal solutions can be computed using a multi-neighborhood tabu search (MNTS) heuristic for the MVWCP.

We also develop an instance generator for the CE problem. 50 randomly generated instances with up to 12,000 bids are tested to evaluate the performance of the GUROBI MIP solver and our Lagrangian relaxation approach. The results have shown that our algorithm significantly outperforms the solver on hard instances, both in solution quality and computation time.

Future research may include the bid generation problem (BGP) and bid pricing problem (BPP) in order to improve the effectiveness of the CE. Combining our CE with routing problems in carrier collaboration is another challenging issue. Moreover, a fair post-collaboration profit allocation mechanism should also be designed to enhance the CE mechanism.

Conclusions and Perspectives

Contents

| | |
|-----------------------------------|------------|
| 6.1 Conclusions | 121 |
| 6.2 Perspectives | 123 |

6.1 Conclusions

In recent years, collaborative logistics or more particularly collaborative transportation has been emerging as an effective strategy for small to medium-sized freight carriers to improve their profitability by reducing empty vehicle repositions and increasing vehicle fill rates. In this thesis, we mainly address collaborative transportation planning (CTP) problems appeared in carrier collaboration, especially the carrier collaboration in less-than-truckload (LTL) transportation.

Two key issues are often addressed for carrier collaboration in LTL transportation : the requests reassignment problem and the post-collaboration profit allocation problem. We focus on the requests reassignment problem in this thesis, and propose an auction-based carrier collaboration framework. In the framework, three main sub-problems have been addressed : the pickup and delivery problem with time windows, profits,

and reserved requests (PDPTWPR), the stochastic bid generation problem (BGP) and the winner determination problem (WDP) in carrier collaboration via combinatorial exchange.

After presenting the research background in Chapter 1 and reviewing the state-of-the-art of the research of collaborative transportation planning in Chapter 2, we propose in Chapter 3 the PDPTWPR, a new vehicle routing problem appeared in LTL carrier collaboration. The PDPTWPR can be considered a request selection problem for a single carrier to determine which requests should be kept to serve by itself and which requests should be outsourced to other carriers in coalition. This is also a preliminary step of the auction process in our carrier collaboration framework. As the solution approach, an adaptive large neighborhood search (ALNS) is developed. Numerical experimental results show that our ALNS gives promising results compared with commercial solver CPLEX.

In Chapter 4, we address a stochastic BGP raised in combinatorial auctions for carrier collaboration in LTL transportation. The BGP aims to aid a carrier to tender bids in order to serve more profitable requests from other carriers in coalition facing uncertain behaviors of bidding of the other carriers. This problem combines request selection problem and routing problem for serving pickup and delivery requests with time windows. A stochastic mathematical programming model is proposed for the BGP and is equivalently transformed into a deterministic one, which can be decomposed into independent pickup and delivery problems with time windows. A GRASP \times ILS heuristic is developed for the resolution of the decomposed sub-problems.

In Chapter 5, we propose a combinatorial exchange (CE) framework for carrier collaboration, which is an alternative mechanism different from

the traditional combinatorial auctions for optimally reallocating requests among carriers. Each carrier plays a double role of seller and buyer of transportation requests in CE. This property confers some advantages to CE over combinatorial auctions. A Lagrangian relaxation approach is developed to solve the winner determination problem in CE.

In summary, the main contributions of this thesis include three new models appeared in CTP of carrier collaboration and the development of efficient solution approaches for them : 1) the PDPTWPR, a new vehicle routing problem to identify profitable requests. 2) the CE : a new transaction mechanism to reallocate requests among carriers. 3) the stochastic BGP for the LTL transportation, a new bid generation problem appeared in LTL carrier collaboration.

6.2 Perspectives

Although we have proposed a complete framework to tackle the requests reassignment problem in LTL carrier collaboration, there are still other works to be done to improve the models and the solution approaches in order to make them more applicable in realistic carrier collaboration environments.

Firstly, for the PDPTWPR, the pickup and delivery requests with time windows are classified into two types : reserved requests and selective requests. The reserved requests are compulsory requests (Ziebuhr & Kopfer, 2014, 2016) that must be fulfilled by itself and cannot be subcontracted to other carriers (Schönberger, 2006; Özener et al., 2011). However, in some cases, subcontracting reserved requests may be possible. In other words, more general vehicle routing problems with both profits and subcontrac-

ting option of reserved requests should be considered in order to deal with the realistic situations when reserved requests are involved and subcontracting of transportation requests is possible. For the solution approach of the PDPTWPR, although the ALNS algorithm outperforms the commercial solver (CPLEX), more efficient and effective meta-heuristics need to be developed in order to quickly solve large realistic size instances.

Secondly, for the stochastic BGP, we only consider a limited number of requests for auction since the proposed algorithm must enumerate all the potential bids based on all the requests for auction, which is the power-set of the number of the requests for auction. Obviously, it is not a clever way to explore all the potential bids since such a way is too time-consuming. So some particular techniques to reduce the number of bids while still guarantee the quality of solution should be developed in future research, such as in [Buer \(2014\)](#). Moreover, we assume the price of each request for auction follows a normal distribution, but, more complicated price distributions should be considered to better simulate a free market environment.

Thirdly, for the CE mechanism, each carrier plays a double role of seller and buyer of transportation requests. The idea makes the tendered bids to have two parts (requests to sell and requests to buy) instead of one in traditional combinatorial auctions (requests to buy). However, there are many ways to determine requests to buy, such as the stochastic BGP in this thesis, whereas the methods to identify requests to sell remain scarce in the literature.

Finally, profit allocation techniques are not addressed in this thesis. But this issue is inevitable when implementing collaborative logistics. In the literature, the requests reassignment problem and the profit allocation problem are often be handled separately. It would be better if the two pro-

blems could be considered together so that a better outcome can be gained compared to deal with them successively. Furthermore, more sophisticated forms of collaboration may be developed to enrich collaborative logistics. For instance, shippers, carriers, or even customers can be involved together in collaboration. For such kind of collaboration, new models and solution approaches are required. All of the above mentioned issues will be the topics for future research.

Appendix of French Abstract :

Contents

| | |
|--|-----|
| A.1 Introduction générale | 127 |
| A.2 Introduction | 131 |
| A.3 Le problème de ramassage et de livraison avec fenêtres de temps, profits et demandes réservées | 134 |
| A.4 Le problème de génération d'enchère dans le transport de chargement partiel | 140 |
| A.5 Le problème de détermination de gagnants dans la collabo- ration entre transporteurs via l'échange combinatoire | 143 |
| A.6 Conclusion et perspective | 146 |
| References | 149 |

A.1 Introduction générale

La collaboration entre les petites et moyennes entreprises (PME) joue un rôle croissant dans leur fonctionnement/gestion quotidienne. La participa-

tion à un réseau et la collaboration avec d'autres entreprises est devenue une stratégie principale pour eux d'acquérir des avantages concurrentiels dans un environnement sévère. Pour atteindre des économies d'échelle, de plus en plus de PME de transport ont formé des réseaux de collaboration en partageant les tâches et les moyens de transport, afin de réduire les coûts, d'améliorer la réactivité à l'évolution du marché, et de saisir plus d'opportunités commerciales.

La concurrence féroce sur les marchés mondiaux, l'introduction de produits avec des cycles de vie plus courts, l'augmentation des coûts de carburant et des prix du travail, la croissance de la législation du transport et des attentes accrues des clients ont diminué les marges bénéficiaires des transporteurs (Crujssen et al., 2007). Ainsi, comme une stratégie efficace pour les petites ou moyennes transporteurs afin d'améliorer la rentabilité en réduisant les ré-positionnements à vide de véhicules et l'augmentation des taux de remplissage des véhicules, la collaboration entre transporteur est en train d'émerger et d'attirer un intérêt croissant des praticiens industriels et des chercheurs (Dai & Chen, 2009). Certains projets de pilotage mis en œuvre aux États-Unis révèlent que le Transport Collaborative Management (CTM) (y compris la collaboration entre transporteur) peut réduire le kilométrage parcouru par les véhicules vides de 15%, le temps d'attente et le temps de pause des véhicules de 15%, les coûts des conducteurs de 15% et peut augmenter le taux de remplissage de véhicules de 33% (Sutherland, 2009).

La collaboration entre transporteurs se réfère à un partenariat entre plusieurs transporteurs au même niveau d'opérations logistiques. Plusieurs transporteurs forment une coalition et échangent leurs demandes de transport pour explorer une meilleure complémentarité entre les demandes. Cette complémentarité des demandes peut améliorer la planification des tournées tout en augmentant les taux de remplissage des véhicules ou en éliminant les retours vides des véhicules et, par conséquent, réduire leurs coûts de transport. Un problème pour la collaboration entre transporteurs est de savoir comment échanger (réaffecter) de façon optimale les demandes de transport entre les transporteurs afin que leur profit total soit maximisé. L'autre problème est de savoir comment répartir équitablement le bénéfice obtenu grâce à la collaboration entre transporteurs dans une coalition afin de garantir la durabilité de cette alliance (Chen, 2016). Dans cette thèse, nous nous concentrons sur le premier problème qui est également désigné comme problème de planification de transport collaboratif (CTP) (Wang & Kopfer, 2011, 2014; Wang et al., 2014).

Dans cette thèse, nous proposons un cadre pour la collaboration entre transporteur. Dans ce cadre, la collaboration entre transporteurs est réalisée par un échange combinatoire composé d'un commissaire-priseur réel/virtuel et plusieurs transporteurs (enchérisseurs). Chaque transporteur joue un rôle double d'acheteur et de vendeur dans le but d'améliorer sa rentabilité grâce à l'échange de demandes de transport avec d'autres transporteurs. Le commissaire-priseur est responsable de la résolution d'un

problème de détermination des gagnants (WDP) (Ackermann et al., 2011) pour affecter les demandes aux transporteurs gagnants. La séquence d'événements suivante décrit la procédure d'échange de demandes entre transporteurs dans le transport de chargement partiel :

1. Chaque transporteur évalue ses demandes comme rentables ou non rentables en résolvant un problème de ramassage et de livraison avec des fenêtres de temps, profits et demandes réservés (PDPTWPR) (chapitre 3).
2. Demandes non rentables sont soumises au commissaire-priseur en tant que les demandes de sous-traitance.
3. Le commissaire-priseur annonce les demandes de sous-traitance/vente à tous les transporteurs.
4. Chaque transporteur soumet des offres avec prix demandé au commissaire-priseur, et chaque offre comprend une ou plusieurs demandes de transport. Ce problème est appelé le problème de génération d'enchère (BGP) (Triki et al., 2014; Buer, 2014; Kuyzu et al., 2015) (chapitre 4).
5. Le commissaire-priseur résout le WDP afin de réaffecter les demandes en fonction des enchères gagnantes (chapitr 5).

A.2 Introduction

La logistique du fret est spécialisée dans le mouvement (ou renvoi) du fret, d'un endroit à l'autre. Dans la dernière décennie, avec le fourishment du E-commerce et de la mondialisation économique, la logistique du fret a joué un rôle essentiel dans la vie quotidienne activités économiques. Cependant, le développement rapide de la logistique du fret induit une concurrence féroce entre les transporteurs. Pour les petites ou moyennes entreprises (PME) de fret, comment survivre dans un environnement de concurrence constitue un véritable défi. Le défi a suscité la logistique collaborative (CL) ou la gestion collaborative de transport (CTM). CL ou CTM est réalisé grâce à la collaboration horizontale entre plusieurs expéditeurs ou transporteurs soit par le partage des capacités de transport ou des demandes de transport. Après la collaboration, tous les acteurs concernés peuvent améliorer la rentabilité en éliminant retours à vide et en augmentant l'utilisation des véhicules (Dai, 2011). Notons que cette collaboration bénéficie du développement des technologies d'information au cours des dernières années.

La logistique collaborative horizontale fait référence à la collaboration entre plusieurs acteurs au même niveau dans les opérations logistiques telles que la collaboration entre les expéditeurs (fabricants) et la collaboration entre les transporteurs. Deux types de logistique collaborative horizontales sont été étudiés dans la littérature : la collaboration entre expé-

diteurs et la collaboration entre transporteurs. La collaboration entre expéditeurs (Ö. Ergun et al., 2007) considère la situation de collaboration entre un seul transporteur et plusieurs expéditeurs. La collaboration entre les expéditeurs est réalisée par la consolidation de leurs demandes de transport. Grâce à la collaboration, les expéditeurs sont en mesure de réduire leurs «coûts cachés» tels que les coûts de repositionnement de véhicules. Cependant, plus d'attention a été mise sur la collaboration entre transporteurs. Autre que la collaboration entre expéditeurs, la collaboration entre transporteurs (Özener et al., 2011 ; Hernández et al., 2012) se passe entre plusieurs transporteurs et examine comment offrir des possibilités aux transporteurs d'exploiter les synergies dans leurs opérations quotidiennes (telles que dans leurs demandes de transport), de réduire les coûts associés au fonctionnement de leurs flottes, réduire les délais de transport, accroître l'utilisation des véhicules, et d'améliorer les niveaux de service au client (Esper & Williams, 2003).

Un problème de collaboration entre transporteurs est d'échanger (réaffecter) de manière optimale les demandes de transport entre plusieurs transporteurs de sorte que leur profit total est maximisée. Ce problème est aussi appelé la planification collaborative de transport (CTP) (Wang & Kopfer, 2011, 2014 ; Wang et al., 2014). L'autre problème est la répartition équitable du bénéfice post-collaboration acquis grâce à la collaboration entre les transporteurs dans une coalition afin de garantir la pérennité de l'alliance. Ce sujet ne sera pas abordé dans la thèse, nous proposons au lecteur de

lire un article récent sur la répartition de bénéfice pour plus d'informations (Guajardo & Rönnqvist, 2016).

Dans cette thèse, nous nous concentrons sur la CTP pour la collaboration entre transporteurs dans le transport de chargement partiel. Deux types d'approches pour ce problème peuvent être trouvés dans la littérature : des approches de planification centralisée et des approches de planification décentralisée. Dans les approches de planification centralisée, un coordinateur central est responsable pour réaffecter les demandes de transport entre les transporteurs de sorte que leur profit total est maximisé. Au contraire, dans les approches de planification décentralisée, chaque transporteur agit comme un agent autonome, il n'y a pas de coordinateur central pour déterminer un plan de transport global pour tous les transporteurs concernés. En raison de la nature des approches de planification centralisée, les informations sur tous les demandes de transport sont accessibles par chaque transporteur en coalition, alors que les informations commerciales confidentielles ne sont pas partagées entre les transporteurs dans les approches de planification décentralisée. Bien que les méthodes centralisées sont plus performantes que les méthodes décentralisées en termes de profit ou de coût, les approches décentralisées sont plus pratiques dans des applications industrielles. Parce que les transporteurs ne sont pas seulement des partenaires mais aussi des concurrents, ils ne veulent pas divulguer des informations sur les clients et les coûts, à leurs partenaires de coopération (Verdonck et al., 2013).

A.3 Le problème de ramassage et de livraison avec fenêtres de temps, profits et demandes réservées

Ce chapitre traite le problème de ramassage et de livraison avec les fenêtres de temps, profits et demandes réservées (PDPTWPR), un nouveau problème de tournées de véhicules apparu dans la collaboration entre transporteurs réalisée par une enchères combinatoire (CA) ou un échange combinatoire (CE). Le PDPTWPR est un sous-problème clé dans le cadre de collaboration entre transporteurs proposé dans cette thèse.

Dans la collaboration entre transporteurs, plusieurs transporteurs forment une alliance et échangent certaines de leurs demandes de transport. Chaque transporteur possède un ensemble de *demandes réservées* (i.e. les demandes non proposées pour l'échange en CA/CE) et peut demander à servir d'autres demandes (*demandes sélectives*) auprès d'autres transporteurs. Chaque demande de transport est une demande de ramassage et de livraison associée à une origine, une destination, une quantité, deux fenêtres de temps et un prix pour servir la demande payé par son expéditeur correspondant (client). Pour chaque transporteur dans CA/CE, il doit déterminer quelles *demandes sélectives* pour enchérir, en plus de ses *demandes réservées*, et construit des tournées réalisables pour maximiser son profit total. Un tel problème soulève une nouvelle variante de problème de ra-

massage et de livraison avec les fenêtres de temps (PDPTW), c'est-à-dire le PDPTWPR. À notre connaissance, ce problème a rarement été étudié dans la littérature.

Un modèle de programmation linéaire en nombres mixtes (MIP) est formulé pour le PDPTWPR et une version améliorée de l'approche de la recherche adaptative à grand voisinage ALNS est développée. L'ALNS implique des opérateurs de destruction/réparation ad hoc et une procédure de post-optimisation menée par recherche locale (LS). L'algorithme s'exécute en segments successifs qui modifient le comportement des opérateurs et calculent leurs propres statistiques pour mettre à jour de manière adaptative les probabilités de sélection des opérateurs. Le modèle MIP et l'approche ALNS sont évalués sur 54 instances générées au hasard avec jusqu'à 100 demandes de transport en charge partielle. Les résultats numériques indiquent que l'ALNS est plus performante de manière significative que le solveur CPLEX, non seulement en termes de qualité de la solution, mais aussi en termes de temps de calcul.

Les contributions de ce chapitre sont résumées comme suit :

1. Le PDPTWPR, une nouvelle variante du problème de tournées de véhicules, est proposé dans le contexte de la collaboration entre transporteurs.
2. Un ensemble d'opérateurs de destruction et de réparation est conçu pour l'algorithme ALNS selon la propriété du PDPTWPR.

3. Un mécanisme qui peut ajuster dynamiquement le comportement des opérateurs pour être conservateur/agressif est utilisé pour le problème de tournées de véhicules avec profits pour la première fois.

Le PDPTWPR est basé sur un graphe complet non orienté $G = (N, E)$. L'ensemble de noeuds du graphe est défini comme $N = \{0, \dots, 2n + 1\}$, où n désigne le nombre de demandes de transport. Les noeuds 0 et $2n + 1$ représentent le dépôt du transporteur, hébergeant un ensemble $K = \{1, \dots, m\}$ de m véhicules identiques de capacité Q . On suppose que chaque tournée de véhicule commence au noeud 0 et se termine au noeud $2n + 1$. Chaque noeud i a une fenêtre de temps $[a_i, b_i]$ pour commencer le service, alors que chaque arc (i, j) dans E est associé à un coût de déplacement c_{ij} et un temps de trajet t_{ij} . Le temps de service au noeud i est inclus dans t_{ij} . Comme dans le VRPTW, un véhicule peut attendre au client i s'il y arrive avant a_i . Le sous-ensemble $P = \{1, \dots, n\}$ contient les noeuds de ramassage de toutes les demandes, tandis que $D = \{n + 1, \dots, 2n\}$ rassemble les noeuds de livraison. La demande $i, i = 1, \dots, n$, est associée à un noeud de ramassage i , un noeud de livraison $n + i$, une demande $d_i > 0$ et un prix p_i . Pour le noeud de livraison, nous définissons $d_{n+i} = -d_i$. L'ensemble R de toutes les demandes comprend le sous-ensemble de demandes réservées R_r et le sous-ensemble de demandes sélectives R_s .

L'objectif du PDPTWPR est de déterminer les demandes sélectives à servir, en plus des demandes réservées, et de déterminer les tournées de véhicules associés, afin de maximiser le profit total qui est égal à la somme des

paiements collectés moins le coût total des tournées. Les demandes servies dans une tournée ne peuvent pas dépasser la capacité du véhicule, la fenêtre de temps à chaque nœud doit être respectée et le nœud de livraison de chaque demande doit être visité après son nœud de ramassage correspondant, dans la même tournée.

Le problème est NP-hard au sens fort comme le PDPTW qui est le cas particulier du problème où R_s est vide et tous les prix p_i sont égaux à une grande constante positive M (pour s'assurer que toutes les demandes sont servies).

Le PDPTWPR peut être formulé par un modèle MIP. En plus des données précédentes, nous avons besoin de deux notations pour formuler le modèle plus facilement : $T_{ij} = b_j - a_i$ joue le rôle d'une constante big-M dans les contraintes de fenêtre de temps, tandis que $Q_i = Q + d_i$ est utilisé dans les contraintes de capacité. Les variables de décision suivantes sont également utilisées dans le modèle :

- x_{ij}^k , la variable binaire est égale à 1 si et seulement si le véhicule k parcourt directement l'arc (i, j) ,
- y_i^k , la variable binaire est égale à 1 si et seulement si la demande i est servie par le véhicule k ,
- T_i^k , heure à laquelle le véhicule k commence le service au nœud i ,
- Q_i^k , charge du véhicule k au départ du nœud i .

Modèle résultant :

$$\max \sum_{k \in K} \sum_{i \in R} p_i \cdot y_i^k - \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} \cdot x_{ij}^k \quad (\text{A.1})$$

Soumis à :

$$\sum_{j \in N, j \neq i} x_{ji}^k - \sum_{j \in N, j \neq i} x_{ij}^k = 0 \quad \forall i \in P \cup D, \forall k \in K \quad (\text{A.2})$$

$$\sum_{j \in P, j \neq 0} x_{0j}^k = 1 \quad \forall k \in K \quad (\text{A.3})$$

$$\sum_{i \in D, i \neq 2n+1} x_{i, 2n+1}^k = 1 \quad \forall k \in K \quad (\text{A.4})$$

$$\sum_{k \in K} y_i^k = 1 \quad \forall i \in R_r \quad (\text{A.5})$$

$$\sum_{k \in K} y_i^k \leq 1 \quad \forall i \in R_s \quad (\text{A.6})$$

$$\sum_{j \in N, j \neq i, 2n+1} x_{ij}^k = y_i^k \quad \forall i \in P, \forall k \in K \quad (\text{A.7})$$

$$\sum_{j \in N, j \neq i, 0} x_{j, n+i}^k = y_i^k \quad \forall i \in P, \forall k \in K \quad (\text{A.8})$$

$$T_i^k + t_{i, n+i} \leq T_{n+i}^k \quad \forall i \in P, \forall k \in K \quad (\text{A.9})$$

$$T_j^k \geq T_i^k + t_{ij} \cdot x_{ij}^k - T_{ij} \cdot (1 - x_{ij}^k) \quad \forall i, j \in N, \forall k \in K \quad (\text{A.10})$$

$$a_i \leq T_i^k \leq b_i \quad \forall i \in N, \forall k \in K \quad (\text{A.11})$$

$$Q_j^k \geq Q_i^k + d_j - Q_j \cdot (1 - x_{ij}^k) \quad \forall i, j \in N, \forall k \in K \quad (\text{A.12})$$

$$\max\{0, d_i\} \leq Q_i^k \leq \min\{Q, Q + d_i\} \quad \forall i \in N, \forall k \in K \quad (\text{A.13})$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in N, \forall k \in K \quad (\text{A.14})$$

$$y_i^k \in \{0, 1\} \quad \forall i \in R, \forall k \in K \quad (\text{A.15})$$

$$T_i^k \geq 0 \quad \forall i \in N, \forall k \in K \quad (\text{A.16})$$

$$Q_i^k \geq 0 \quad \forall i \in N, \forall k \in K \quad (\text{A.17})$$

La fonction objectif (A.1) représente le profit total du transporteur, égale à la différence entre la somme des paiements des demandes servies et le coût total de transport. Les contraintes (A.2) garantissent qu'un véhicule arrivant à un nœud de ramassage ou de livraison doit le quitter. Les contraintes (A.3) et (A.4) indiquent que chaque véhicule quitte le dépôt et y revient. Les contraintes (A.5) garantissent que toutes les demandes réservées doivent être servies une seule fois, tandis que dans les contraintes (A.6) les demandes sélectives sont servies au maximum une fois. Les contraintes (A.7) et (A.8) garantissent que si une demande est sélectionnée pour service, il doit y avoir un véhicule quittant son nœud de ramassage et arrivant à son nœud de livraison correspondant. Les fenêtres de temps et les relations de précedence sont respectées via des contraintes (A.9) à (A.11). Les contraintes (A.12) et (A.13) concernent la capacité de chaque véhicule. Enfin, les contraintes (A.14) à (A.17) définissent les natures des variables.

A.4 Le problème de génération d'enchère dans le transport de chargement partiel

Ce chapitre traite un problème de génération d'enchère stochastique (BGP) soulevé dans des ventes aux enchères combinatoires (CA) pour la collaboration entre transporteurs en mode de chargement partiel (LTL), qui combine le problème de sélection de demandes et le problème de tournées de véhicules pour servir des demandes de ramassage et de livraison avec fenêtres de temps. Dans ce problème, plusieurs transporteurs forment une alliance et échanger leurs demandes de transport dans le but d'améliorer leur rentabilité. Chaque transporteur veut générer la meilleure offre de demandes à l'enchère tout en servant ses demandes réservées avec des fenêtres de temps. Une offre représente un paquet de demandes à servir avec un «prix demandé». Le «prix demandé» indique combien un transporteur charge le commissaire-priseur pour servir le paquet de demandes. Généralement, le «prix demandé» doit être une valeur positive, mais pas être supérieur à la somme du coût de service individuel de chaque demande dans ce paquet. La détermination du «prix demandé» est un problème difficile dans le BGP. D'une part, plus un transporteur demande au commissaire-priseur pour servir un paquet de demandes, plus le profit il peut générer plus tard. D'autre part, les concurrents (autres transporteurs en coalition) peuvent demander des prix plus bas pour servir le même paquet de demandes, en vertu d'une telle circonstance, une offre perdante ne reçoit rien en retour.

Au-delà, nous nous référons à la détermination du «prix demandé» comme le problème de prix.

Triki et al. (2014) a souligné deux défis principaux pour le BGP en TL transport, l'un est la nécessité d'énumérer un nombre exponentiel d'offres potentielles qui est le pouvoir-ensemble de toutes les demandes d'enchères. Cette difficulté fait aussi le BGP un problème NP-hard. Le second problème est la nécessité de considérer les synergies entre les demandes dans une offre en raison de la complémentarité entre elles. L'estimation du facteur de synergie joue un rôle clé dans le contexte du transport LTL, qui sera présenté plus tard avec une description détaillée dans la section 4.2.4.

En règle générale, dans la logistique collaborative, plusieurs acteurs (transporteurs ou expéditeurs) engager dans une procédure de génération d'offres. Par conséquent, les comportements des autres acteurs devraient être pris en considération lorsque nous essayons de résoudre le BGP du point de vue d'un transporteur. Toutefois, les comportements d'autres acteurs sont parfois imprévisibles, même inconnus en raison de la protection d'information commerciale. Donc, il est nécessaire de considérer le facteur stochastique dans le processus BGP pour mieux simuler un environnement de marché libre.

Au mieux de notre connaissance, l'étude du BGP stochastique des ventes de demandes de transport aux enchères combinatoires ou aux enchères simultanées sont limitées en mode de TL. Mais le BGP stochastique dans le transport de mode LTL n'a jamais été abordé auparavant dans la

littérature, et le transport de LTL aussi joue un rôle important dans les opérations de transport de nos jours. Donc, dans ce chapitre, nous essayons de combler cette lacune en proposant un modèle et en élaborant un algorithme heuristique efficace. Motivé par le modèle d'optimisation probabiliste dans le transport de TL proposé par [Triki et al. \(2014\)](#), nous étendons et l'appliquons au transport de LTL. Le modèle de programmation stochastique quadratique en nombres mixtes (MIQP) est ensuite transformé en plusieurs problèmes de ramassage et de livraison indépendants avec des fenêtres de temps (PDPTW). Un GRASP \times ILS algorithme est développé en tant que l'approche de résolution du problème. Des expérimentations numériques sur des instances générées aléatoirement montrent l'efficacité de l'algorithme.

Les contributions de ce chapitre sont résumées comme suit :

1. Le BGP stochastique pour le transport LTL est étudié pour la première fois.
2. Le modèle stochastique MIQP du problème est simplifié, transformé et décomposé en plusieurs PDPTW indépendants.
3. Une technique pertinentes est proposée pour estimer le facteur de synergie entre les demandes de transport, fondée sur les comportements d'autres transporteurs (concurrents).

A.5 Le problème de détermination de gagnants dans la collaboration entre transporteurs via l'échange combinatoire

Selon un rapport autorisé fourni par Eurostat en 2013 (économie d'entreprise-analyse de classe de taille), les PME a gagné 54% de la part de marché de l'UE dans le secteur de transport et de stockage¹. Par conséquent, les stratégies d'opérations plus efficaces pour améliorer la compétitivité et de gagner plus de profit pour les PME dans le transport de marchandises/secteur de la logistique ont été largement sollicitées au cours de la dernière décennie. La logistique collaborative (CL) est l'une de ces stratégies. Comme un moyen efficace pour réduire les retours à vide de véhicules et d'augmenter leurs taux d'utilisation, CL a attiré un intérêt croissant de praticiens industriels et de chercheurs universitaires (Dai & Chen, 2009). Au point de vue économique et écologique, CL est en effet un moyen essentiel pour réduire les distances parcourues par les véhicules, en particulier pour éviter deadheads, *i.e.*, les déplacements de véhicules vides (Ackermann et al., 2011).

En général, CL doit faire face à deux sous-problèmes, *i.e.*, la réaffectation des demandes de transport et le partage de profit parmi les transporteurs (Krajewska & Kopfer, 2006; Berger & Bierwirth, 2010; Dai & Chen,

1. http://ec.europa.eu/eurostat/statistics-explained/index.php/Business_economy_-_size_class_analysis

2011; Robu et al., 2011; Dai et al., 2014; Wang & Kopfer, 2014; Wang et al., 2014; Dai & Chen, 2015; Y. Li et al., 2016). Dans ce chapitre, nous nous concentrons sur le premier sous-problème : la conception d'un mécanisme d'affectation des demandes de transport entre les transporteurs dans un échange combinatoire (CE). Ce mécanisme du CE considère un scénario où chaque transporteur répond à un appel d'offres pour l'acquisition (l'achat) des demandes d'autres transporteurs et la vente de certaines de ses demandes à d'autres transporteurs en même temps afin de maximiser son profit. Dans le CE, les transporteurs soumettent toutes leurs demandes de sous-traitance à un commissaire-priseur virtuel, puis les demandes sont ré-affectés parmi eux par le commissaire-priseur, selon toutes les offres reçues. La caractéristique bi-directionnelle du CE, le rendre en mesure d'exploiter des profits potentiels par une meilleure exploration des synergies entre les demandes et par une plus grande participation des transporteurs. Au mieux de notre connaissance, ce sujet n'a qu'été conceptuellement abordé (Bloos & Kopfer, 2009; Ackermann et al., 2011) mais jamais étudié profondément dans la littérature.

Le CE est un mécanisme de transaction alternatif aux ventes aux enchères combinatoires (CA) entre les transporteurs de LTL. Dans le CE, chaque transporteur joue un rôle double d'acheteur et de vendeur. Cette caractéristique apporte certains avantages par rapport aux mécanismes de CA traditionnels. Un modèle de programmation linéaire binaire est formulé pour le problème. Une approche de relaxation lagrangienne est en-

suite développée pour résoudre le problème du CE. Le problème relaxé se transforme en un problème de clique pondéré avec la maximisation du poids total des noeuds (MVWCP) qui est résolu soit par une méthode de recherche multi-voisinages tabu (MNTS) ou par un solveur commercial de MIP. Une heuristique de réparation est proposée pour construire une solution faisable à partir d'une solution infaisable dont les infaisabilités causées par la relaxation de contraintes. Les expérimentations numériques intensives sur les instances générées aléatoirement montrent que l'approche de relaxation lagrangienne peut fournir des solutions de haute qualité.

Les contributions de ce chapitre sont résumées comme suit :

1. Nous introduisons un mécanisme de CE comme une alternative à la CA pour la réaffectation/échange de demandes de transport entre transporteurs dans le transport de LTL.
2. Un modèle mathématique qui peut être adapté à différentes langues d'appel d'offres est élaboré pour le problème du CE.
3. Comme l'approche de résolution pour le problème du CE, une approche efficace basée sur la relaxation lagrangienne et l'algorithme MNTS pour résoudre le problème clique pondéré est développée et prouvée capable de produire des solutions de haute qualité par des expérimentations numériques sur des instances générées aléatoirement.

A.6 Conclusion et perspective

Bien que nous ayons proposé un cadre complet pour aborder le problème de réaffectation des demandes de transport parmi les transporteurs de chargement partiel en collaboration, il reste encore beaucoup de travail à faire pour améliorer les modèles et les approches de résolution en vue d'adapter à des problèmes de la logistique collaborative plus généraux et à des problèmes de grande taille dans des applications industrielles.

Tout d'abord, pour le PDPTWPR, nous ont classifié toutes les demandes de transport de ramassage et de livraison avec des fenêtres de temps en deux types : demandes réservées et demandes sélectives. Des demandes réservées sont des demandes obligatoires qui doivent être servies par un transporteur lui-même (Ziebuhr & Kopfer, 2014, 2016) et qui ne peuvent pas être sous-traitées à d'autres transporteurs (Schönberger, 2006; Özener et al., 2011). Mais dans certains cas, il n'y a pas de demande strictement réservée, parce que toute violation du contrat de servir une telle demande peut être compensée par une pénalité. En d'autres termes, les modèles de problèmes de tournée des véhicules plus généraux avec pénalités doivent être considérés afin d'adapter à une situation réelle lorsque les demandes réservées sont impliquées. Pour l'approche de résolution du PDPTWPR, bien que l'algorithme ALNS est plus performant que le solveur commercial (CPLEX), des méta-heuristiques plus efficaces doivent être développées afin de résoudre des instances de la taille de l'industrie plus rapidement.

Deuxièmement, pour le problème de génération d'enchère dans le transport de chargement partiel, nous ne considérons qu'un nombre limité de demandes aux enchères puisque l'algorithme proposé doit énumérer toutes les offres potentielles sur la base de toutes les demandes aux enchères, qui est la puissance du nombre des demandes. Evidemment, ce n'est pas une manière intelligente d'explorer toutes les offres potentielles puisque une telle manière prend trop de temps. Ainsi, des techniques particulières pour réduire le nombre d'offres tout en garantissant la qualité de solution devraient être développées pour les recherches futures, comme fait dans [Buer \(2014\)](#). En outre, nous supposons que les prix des demandes de transport aux enchères suivent une distribution normale, mais une distribution de prix plus compliquée devrait être envisagée pour mieux simuler un environnement de marché libre. Les deux aspects ci-dessus peuvent être les directions de recherche future.

Troisièmement, pour le mécanisme d'échange combinatoire de demandes de transport en charge partielle, chaque transporteur joue un double rôle de vendeur et d'acheteur. L'idée est que les appels d'offres concernent deux parties (demandes pour la vente et demandes pour l'achat) au lieu d'une partie dans les enchères combinatoires traditionnelles (demandes de l'achat). Cependant, il existe de nombreuses façons de déterminer les demandes pour l'achat, comme le problème de génération d'enchère stochastique dans cette thèse, alors que les méthodes pour identifier les demandes pour la vente restent rares dans la littérature.

Enfin, les techniques de répartition de profit ne sont pas abordées dans cette thèse. Mais ce problème est important lorsque l'on considère la logistique collaborative. Dans la littérature, le problème de la réaffectation des demandes de transport et le problème de l'allocation de profit sont souvent traités séparément. Nous pensons que des techniques peuvent être proposées pour intégrer les deux problèmes ensemble de sorte qu'un meilleur résultat peut être obtenu par rapport au traitement des deux problèmes successivement. En outre, des formes plus sophistiquées de collaboration peuvent être envisagées dans le domaine de la logistique collaborative, par exemple lorsque les expéditeurs, ou même les clients, sont impliqués ensemble dans la collaboration. Pour ce type de collaboration, de nouveaux modèles et des approches de résolution sont exigés. Tous ces problèmes ci-dessus seront les sujets de la recherche future.

References

- Ackermann, H., Ewe, H., Kopfer, H., & Küfer, K.-H. (2011). Combinatorial auctions in freight logistics. In *International conference on computational logistics* (pp. 1–17). (Cited on pages 2, 17, 89, 93, 95, 130, 143 and 144.)
- Aksen, D., Kaya, O., Salman, F. S., & Tüncel, Ö. (2014). An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. *European Journal of Operational Research*, 239(2), 413–426. (Cited on pages 32 and 33.)
- An, N., Elmaghraby, W., & Keskinocak, P. (2005). Bidding strategies and their impact on revenues in combinatorial auctions. *Journal of Revenue and Pricing Management*, 3(4), 337–357. (Cited on page 72.)
- Andersson, A., Tenhunen, M., & Ygge, F. (2000). Integer programming for combinatorial auction winner determination. In *Multiagent systems, 2000. proceedings. fourth international conference on* (pp. 39–46). (Cited on page 94.)
- Archetti, C., Corberan, A., Sanchis, J., Plana, I., & Speranza, M. (2014). The team orienteering arc routing problem. *Transportation Science*, 48, 442–457. (Cited on page 27.)
- Archetti, C., Speranza, M., & Vigo, D. (2013). *Vehicle routing problems with profits* (Rapport technique). Tech. Report WPDEM2013/3, University of Brescia. (Cited on page 28.)
- Ausiello, G., D’Atri, A., & Protasi, M. (1980). Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences*, 21(1), 136–153. (Cited on page 104.)
- Babaioff, M., & Walsh, W. E. (2005). Incentive-compatible, budget-balanced, yet highly efficient auctions for supply chain formation. *Decision Support Systems*, 39(1), 123–149. (Cited on page 16.)
- Babel, L. (1994). A fast algorithm for the maximum weight clique problem. *Computing*, 52(1), 31–38. (Cited on page 92.)
- Balas, E. (1989). The prize collecting traveling salesman problem. *Networks*, 19(6), 621–636. (Cited on page 27.)
- Balas, E., & Yu, C. S. (1986). Finding a maximum clique in an arbitrary graph. *SIAM Journal on Computing*, 15(4), 1054–1068. (Cited on page 92.)
- Benlic, U., & Hao, J.-K. (2013). Breakout local search for maximum clique problems. *Computers & Operations Research*, 40(1), 192–206. (Cited on pages 92 and 93.)
- Bent, R., & Hentenryck, P. V. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33(4), 875–893. (Cited on pages 26 and 31.)

- Berger, S., & Bierwirth, C. (2010). Solutions to the request reassignment problem in collaborative carrier networks. *Transportation Research Part E : Logistics and Transportation Review*, 46(5), 627–638. (Cited on pages 16, 88 and 143.)
- Bloos, M., & Kopfer, H. (2009). Efficiency of transport collaboration mechanisms. *Communications of SIWN*, 6, 23–28. (Cited on pages 89 and 144.)
- Bomze, I. M., Pelillo, M., & Stix, V. (2000). Approximating the maximum weight clique using replicator dynamics. *IEEE Transactions on Neural Networks*, 11(6), 1228–1241. (Cited on page 93.)
- Boughaci, D., Benhamou, B., & Drias, H. (2009). A memetic algorithm for the optimal winner determination problem. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 13(8), 905–917. (Cited on page 91.)
- Boughaci, D., Benhamou, B., & Drias, H. (2010). Local search methods for the optimal winner determination problem in combinatorial auctions. *Journal of Mathematical Modelling and Algorithms*, 9(2), 165–180. (Cited on page 91.)
- Boutilier, C., & Hoos, H. H. (2001). Bidding languages for combinatorial auctions. In *International joint conference on artificial intelligence* (Vol. 17, pp. 1211–1217). (Cited on page 99.)
- Buer, T. (2014). An exact and two heuristic strategies for truthful bidding in combinatorial transport auctions. *arXiv preprint arXiv :1406.1928*. (Cited on pages 3, 21, 68, 96, 124, 130 and 147.)
- Busygin, S. (2006). A new trust region technique for the maximum weight clique problem. *Discrete Applied Mathematics*, 154(15), 2080–2096. (Cited on page 93.)
- Butt, S. E., & Cavalier, T. M. (1994). A heuristic for the multiple tour maximum collection problem. *Computers & Operations Research*, 21(1), 101–111. (Cited on page 27.)
- Chang, T.-S. (2009). Decision support for truckload carriers in one-shot combinatorial auctions. *Transportation Research Part B : Methodological*, 43(5), 522–541. (Cited on page 72.)
- Chao, I., Golden, B. L., & Wasil, E. A. (1996). The team orienteering problem. *European Journal of Operational Research*, 88(3), 464–474. (Cited on page 27.)
- Chen, H. (2016). Combinatorial clock-proxy exchange for carrier collaboration in less than truck load transportation. *Transportation Research Part E : Logistics and Transportation Review*, 91, 152–172. (Cited on pages 2, 19 and 129.)
- Cramton, P., Shoham, Y., & Steinberg, R. (2006). Combinatorial auctions. (Cited on page 14.)
- Cruijssen, F., Cools, M., & Dullaert, W. (2007). Horizontal cooperation in logistics : opportunities and impediments. *Transportation Research*

- Part E : Logistics and Transportation Review*, 43(2), 129–142. (Cited on pages 1, 8, 10 and 128.)
- Dai, B. (2011). *Collaborative logistics planning among carriers (models and approaches)* (Thèse de doctorat non publiée). (Cited on pages 8 and 131.)
- Dai, B., & Chen, H. (2009). Mathematical model and solution approach for collaborative logistics in less than truckload (LTL) transportation. In *International conference on computers & industrial engineering (cie 2009)* (pp. 767–772). (Cited on pages 1, 128 and 143.)
- Dai, B., & Chen, H. (2011). A multi-agent and auction-based framework and approach for carrier collaboration. *Logistics Research*, 3(2-3), 101–120. (Cited on pages 18, 88, 143 and 144.)
- Dai, B., & Chen, H. (2015). Proportional egalitarian core solution for profit allocation games with an application to collaborative transportation planning. *European Journal of Industrial Engineering*, 9(1), 53–76. (Cited on pages 88 and 144.)
- Dai, B., Chen, H., & Yang, G. (2014). Price-setting based combinatorial auction approach for carrier collaboration with pickup and delivery requests. *Operational Research*, 14(3), 361–386. (Cited on pages 19, 88 and 144.)
- D'Amours, S., & Rönnqvist, M. (2010). Issues in collaborative logistics. In *Energy, natural resources and environmental economics* (pp. 395–409). Springer. (Cited on page 11.)
- Day, R. W. (2004). Expressing preferences with price-vector agents in combinatorial auctions. (Cited on page 16.)
- De Vries, S., & Vohra, R. V. (2003). Combinatorial auctions : A survey. *INFORMS Journal on computing*, 15(3), 284–309. (Cited on page 15.)
- Dumas, Y., Desrosiers, J., & Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54(1), 7–22. (Cited on page 63.)
- Ergun, O., Kuyzu, G., & Savelsbergh, M. (2003). The lane covering problem. *Manuscript*. (Cited on page 12.)
- Ergun, O., Kuyzu, G., & Savelsbergh, M. (2007a). *Bid price optimization for simultaneous truckload transportation procurement auctions* (Rapport technique). Citeseer. (Cited on page 21.)
- Ergun, O., Kuyzu, G., & Savelsbergh, M. (2007b). Reducing truckload transportation costs through collaboration. *Transportation Science*, 41(2), 206–221. (Cited on pages 12 and 13.)
- Ergun, Ö., Kuyzu, G., & Savelsbergh, M. (2007). Shipper collaboration. *Computers & Operations Research*, 34(6), 1551–1560. (Cited on pages 8, 12 and 132.)
- Escudero, L. F., Landete, M., & Marín, A. (2009). A branch-and-cut algorithm for the winner determination problem. *Decision Support Sys-*

- tems*, 46(3), 649–659. (Cited on page 91.)
- Esper, T. L., & Williams, L. R. (2003). The value of collaborative transportation management (ctm) : its relationship to cpfr and information technology. *Transportation Journal*, 55–65. (Cited on pages 9 and 132.)
- Feillet, D., Dejax, P., & Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation Science*, 39(2), 188–205. (Cited on page 27.)
- Figliozzi, M. (2006). Analysis and evaluation of incentive-compatible dynamic mechanisms for carrier collaboration. *Transportation Research Record : Journal of the Transportation Research Board*(1966), 34–40. (Cited on page 15.)
- Frieze, A. M., Galbiati, G., & Maffioli, F. (1982). On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 12(1), 23–39. (Cited on page 27.)
- Fujishima, Y., Leyton-Brown, K., & Shoham, Y. (1999). Taming the computational complexity of combinatorial auctions : Optimal and approximate approaches. In *Ijcai* (Vol. 99, pp. 548–553). (Cited on page 91.)
- Gibbons, R. (1992). *A primer in game theory*. Published by the author. (Cited on page 95.)
- Goemans, M. X., & Bertsimas, D. J. (1990). On the parsimonious property of connectivity problems. In *Proceedings of the first annual acm-siam symposium on discrete algorithms* (pp. 388–396). (Cited on page 27.)
- Goossens, D., & Spieksma, F. (2007). The matrix bid auction : micro-economic properties and expressiveness. Available at SSRN 968583. (Cited on page 16.)
- Guajardo, M., & Rönnqvist, M. (2016). A review on cost allocation methods in collaborative transportation. *International Transactions in Operational Research*, 23(3), 371–392. (Cited on pages 9 and 133.)
- Günlük, O., Ladányi, L., & De Vries, S. (2005). A branch-and-price algorithm and new test problems for spectrum auctions. *Management Science*, 51(3), 391–406. (Cited on page 91.)
- Guo, Y., Lim, A., Rodrigues, B., & Tang, J. (2006). Using a lagrangian heuristic for a combinatorial auction problem. *International Journal on Artificial Intelligence Tools*, 15(03), 481–489. (Cited on page 91.)
- Handoko, S. D., & Lau, H. C. (2016). Enabling carrier collaboration via order sharing double auction : A singapore urban logistics perspective. *Transportation Research Procedia*, 12, 777–786. (Cited on page 17.)
- Hernández, S., & Peeta, S. (2014). A carrier collaboration problem for less-than-truckload carriers : characteristics and carrier collaboration model. *Transportmetrica A : Transport Science*, 10(4), 327–349. (Cited on page 13.)
- Hernández, S., Unnikrishnan, A., & Awale, S. (2012). Centralized carrier collaboration multihub location problem for less-than-truckload industry : Hybrid hub-and-spoke network. *Transportation Research Re-*

- cord : *Journal of the Transportation Research Board*(2269), 20–28. (Cited on pages 8 and 132.)
- Hoos, H. H., & Boutilier, C. (2000). Solving combinatorial auctions using stochastic local search. In *Aaai/iaai* (pp. 22–29). (Cited on page 91.)
- Hosny, M. I., & Mumford, C. L. (2012). Constructing initial solutions for the multiple vehicle pickup and delivery problem with time windows. *Journal of King Saud University-Computer and Information Sciences*, 24(1), 59–69. (Cited on page 26.)
- Hsieh, F.-S. (2010). Combinatorial reverse auction based on revelation of Lagrangian multipliers. *Decision Support Systems*, 48(2), 323–330. (Cited on page 91.)
- Hsieh, F.-S., & Tsai, S.-M. (2008). Combinatorial reverse auction based on Lagrangian relaxation. In *Ieee asia-pacific services computing conference (apsc'08)* (pp. 329–334). (Cited on page 91.)
- Kameshwaran, S., Benyoucef, L., & Xie, X. (2005). Design of progressive auctions for procurement based on Lagrangian relaxation. In *Ieee int. conf. on e-commerce technology (cec 2005)* (pp. 9–16). (Cited on page 91.)
- Kopfer, H., & Krajewska, M. A. (2007). Approaches for modelling and solving the integrated transportation and forwarding problem. *Produktions-und Logistikmanagement. Verlag Franz Vahlen, Munich*, 439–458. (Cited on page 11.)
- Krajewska, M. A., & Kopfer, H. (2006). Collaborating freight forwarding enterprises. *OR spectrum*, 28(3), 301–317. (Cited on pages 16, 88 and 143.)
- Krishna, V. (2009). *Auction theory*. Academic press. (Cited on page 14.)
- Kuyzu, G., Akyol, Ç. G., Ergun, Ö., & Savelsbergh, M. (2015). Bid price optimization for truckload carriers in simultaneous transportation procurement auctions. *Transportation Research Part B : Methodological*, 73, 34–58. (Cited on pages 3, 21 and 130.)
- Kwon, R. H., Lee, C.-G., & Ma, Z. (2005). An integrated combinatorial auction mechanism for truckload transportation procurement. *University of Toronto*, <http://www.mie.utoronto.ca/labs/ilr/MultiRound.pdf>, Working Paper. (Cited on pages 15 and 17.)
- Labadie, N., Mansini, R., Melechovsky, J., & Wolfler-Calvo, R. (2012). The team orienteering problem with time windows : An LP-based granular variable neighborhood search. *European Journal of Operational Research*, 220, 15–27. (Cited on pages 27 and 31.)
- Lau, H. C., & Goh, Y. G. (2002). An intelligent brokering system to support multi-agent web-based 4 th-party logistics. In *14th ieee international conference on tools with artificial intelligence (ictai 2002)* (pp. 154–161). (Cited on page 118.)
- Lee, C.-G., Kwon, R. H., & Ma, Z. (2007). A carrier's optimal bid generation problem in combinatorial auctions for transportation procurement. *Transportation Research Part E : Logistics and Transportation Re-*

- view*, 43(2), 173–191. (Cited on pages 20 and 72.)
- Leyton-Brown, K., Pearson, M., & Shoham, Y. (2000). Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the 2nd acm conference on electronic commerce* (pp. 66–76). (Cited on page 118.)
- Leyton-Brown, K., Shoham, Y., & Tennenholtz, M. (2000). An algorithm for multi-unit combinatorial auctions. In *Aaai/iaai* (pp. 56–61). (Cited on page 91.)
- Li, H., & Lim, A. (2003). A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools*, 12(02), 173–186. (Cited on page 26.)
- Li, J., Rong, G., & Feng, Y. (2015). Request selection and exchange approach for carrier collaboration based on auction of a single request. *Transportation Research Part E : Logistics and Transportation Review*, 84, 23–39. (Cited on page 19.)
- Li, Y., Chen, H., & Prins, C. (2016). Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. *European Journal of Operational Research*, 252(1), 27–38. (Cited on pages 88 and 144.)
- Mannino, C., & Stefanutti, E. (1999). An augmentation algorithm for the maximum weighted stable set problem. *Computational Optimization and Applications*, 14(3), 367–381. (Cited on page 93.)
- Mansouri, B., & Hassini, E. (2015). A Lagrangian approach to the winner determination problem in iterative combinatorial reverse auctions. *European Journal of Operational Research*, 244(2), 565–575. (Cited on page 91.)
- Massaro, A., Pelillo, M., & Bomze, I. M. (2002). A complementary pivoting approach to the maximum weight clique problem. *SIAM Journal on Optimization*, 12(4), 928–948. (Cited on page 93.)
- Nandiraju, S., & Regan, A. . (2004). Optimization models for auctions for transportation service contract procurement. (Cited on page 91.)
- Nanry, W. P., & Wesley Barnes, J. (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B : Methodological*, 34(2), 107–121. (Cited on page 26.)
- Nguyen, V. H., & Nguyen, T. T. T. (2010). Approximating the asymmetric profitable tour. *Electronic Notes in Discrete Mathematics*, 36, 907–914. (Cited on page 27.)
- Nisan, N. (2000). Bidding and allocation in combinatorial auctions. In *Proceedings of the 2nd acm conference on electronic commerce* (pp. 1–12). (Cited on pages 91 and 99.)
- Nisan, N. (2006). Bidding languages. *Combinatorial auctions*. Cambridge : MIT. (Cited on page 99.)

- Nisan, N. (2007). Introduction to mechanism design (for computer scientists). *Algorithmic game theory*, 209, 242. (Cited on page 16.)
- Ozener, O. O., Ergun, O., & Savelsbergh, M. (2009). Collaboration for truckload carriers. In *Infirms annual conference, san diego, ca.* (Cited on page 13.)
- Özener, O. Ö., Ergun, Ö., & Savelsbergh, M. (2011). Lane-exchange mechanisms for truckload carrier collaboration. *Transportation Science*, 45(1), 1–17. (Cited on pages 8, 123, 132 and 146.)
- Padberg, M. W. (1973). On the facial structure of set packing polyhedra. *Mathematical programming*, 5(1), 199–215. (Cited on page 91.)
- Pardalos, P. M., Migdalas, A., & Pitsoulis, L. (2008). *Pareto optimality, game theory and equilibria* (Vol. 17). Springer Science & Business Media. (Cited on page 15.)
- Pareto, V. (1971). *Manual of political economy* (reprint in English). (Cited on page 15.)
- Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & operations research*, 34(8), 2403–2435. (Cited on page 32.)
- Prins, C. (2009). A grasp× evolutionary local search hybrid for the vehicle routing problem. In *Bio-inspired algorithms for the vehicle routing problem* (pp. 35–53). Springer. (Cited on page 76.)
- Pullan, W. (2008). Approximating the maximum vertex/edge weighted clique using local search. *Journal of Heuristics*, 14(2), 117–134. (Cited on page 93.)
- Robu, V., Noot, H., La Poutré, H., & Van Schijndel, W.-J. (2011). A multi-agent platform for auction-based allocation of loads in transportation logistics. *Expert Systems with Applications*, 38(4), 3483–3491. (Cited on pages 88 and 144.)
- Ropke, S., & Cordeau, J.-F. (2009). Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3), 267–286. (Cited on pages 80 and 112.)
- Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4), 455–472. (Cited on pages 26, 32, 39, 40, 43 and 49.)
- Rothkopf, M. H., Pekeč, A., & Harstad, R. M. (1998). Computationally manageable combinatorial auctions. *Management Science*, 44(8), 1131–1147. (Cited on pages 90 and 91.)
- Sandholm, T. (2002). Algorithm for optimal winner determination in combinatorial auctions. *Artificial intelligence*, 135(1), 1–54. (Cited on page 14.)
- Sandholm, T., & Suri, S. (2003). BOB : Improved winner determination in combinatorial auctions and generalizations. *Artificial Intelligence*,

- 145(1), 33–58. (Cited on page 91.)
- Sandholm, T., Suri, S., Gilpin, A., & Levine, D. (2001). CABOB : A fast optimal algorithm for combinatorial auctions. In *International joint conference on artificial intelligence* (Vol. 17, pp. 1102–1108). (Cited on page 91.)
- Sandholm, T., Suri, S., Gilpin, A., & Levine, D. (2002). Winner determination in combinatorial auction generalizations. In *1st int. joint conf. on autonomous agents and multi-agent systems : part 1* (pp. 69–76). (Cited on page 94.)
- Schönberger, J. (2006). *Operational freight carrier planning : basic concepts, optimization models and advanced memetic algorithms*. Springer Science & Business Media. (Cited on pages 123 and 146.)
- Schwind, M., Gujo, O., & Vykoukal, J. (2009). A combinatorial intra-enterprise exchange for logistics services. *Information systems and e-business management*, 7(4), 447–471. (Cited on page 16.)
- Sghir, I., Hao, J.-K., Jaafar, I. B., & Ghédira, K. (2014). A recombination-based tabu search algorithm for the winner determination problem. In *Artificial evolution* (pp. 157–167). (Cited on page 91.)
- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and practice of constraint programming* (pp. 417–431). Springer. (Cited on pages 31 and 43.)
- Sheffi, Y. (2004). Combinatorial auctions in the procurement of transportation services. *Interfaces*, 34(4), 245–252. (Cited on page 15.)
- Shoham, Y., & Leyton-Brown, K. (2008). *Multiagent systems : Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press. (Cited on page 14.)
- Singh, A., & Gupta, A. K. (2006). A hybrid evolutionary approach to maximum weight clique problem. *International Journal of Computational Intelligence Research*, 2(4), 349–355. (Cited on page 93.)
- Song, J., & Regan, A. (2003b). Combinatorial auctions for transportation service procurement : The carrier perspective. *Transportation Research Record : Journal of the Transportation Research Board*(1833), 40–46. (Cited on page 15.)
- Song, J., & Regan, A. C. (2003a). An auction based collaborative carrier network. *University of California Transportation Center*. (Cited on page 15.)
- Sutherland, J. (2009). *Collaborative transportation management : What, why, and how?* (Rapport technique). Center for Value Chain Research, Lehigh University. Consulté sur <http://docslide.us/business/collaborative-transportation-management.html> (PowerPoint presentation, last accessed on November 13, 2015) (Cited on pages 2 and 128.)
- Toth, P., & Vigo, D. (2014). *Vehicle routing : Problems, methods and applications, second edition*. Philadelphia : SIAM. (Cited on page 45.)

- Triki, C., Oprea, S., Beraldi, P., & Crainic, T. G. (2014). The stochastic bid generation problem in combinatorial transportation auctions. *European Journal of Operational Research*, 236(3), 991–999. (Cited on pages 3, 21, 61, 64, 70, 130, 141 and 142.)
- Verdonck, L., Caris, A., Ramaekers, K., & Janssens, G. K. (2013). Collaborative logistics from the perspective of road transportation companies. *Transport Reviews*, 33(6), 700–719. (Cited on pages 10, 11 and 133.)
- Wang, X., & Kopfer, H. (2011). Increasing efficiency of freight carriers through collaborative transport planning : Chances and challenges. In *Dynamics and sustainability in international logistics and supply chain management-proceedings of the 6th german-russian logistics and scm workshop (dr-log 2011)*. cuvillier verlag, göttingen (pp. 41–50). (Cited on pages 2, 9, 129 and 132.)
- Wang, X., & Kopfer, H. (2014). Collaborative transportation planning of less-than-truckload freight. *OR spectrum*, 36(2), 357–380. (Cited on pages 2, 9, 18, 88, 98, 129, 132 and 144.)
- Wang, X., Kopfer, H., & Gendreau, M. (2014). Operational transportation planning of freight forwarding companies in horizontal coalitions. *European Journal of Operational Research*, 237(3), 1133–1141. (Cited on pages 2, 9, 88, 129, 132 and 144.)
- Wang, X., & Xia, M. (2005). Combinatorial bid generation problem for transportation service procurement. *Transportation Research Record : Journal of the Transportation Research Board*(1923), 189–198. (Cited on pages 20 and 72.)
- Warren, J. S., & Hicks, I. V. (2006). Combinatorial branch-and-bound for the maximum weight independent set problem. *Technical Report, Texas A&M University*. (Cited on page 92.)
- Wu, Q., & Hao, J.-K. (2015a). A review on algorithms for maximum clique problems. *European Journal of Operational Research*, 242(3), 693–709. (Cited on page 92.)
- Wu, Q., & Hao, J.-K. (2015b). Solving the winner determination problem via a weighted maximum clique heuristic. *Expert Systems with Applications*, 42(1), 355–365. (Cited on pages 91 and 104.)
- Wu, Q., Hao, J.-K., & Glover, F. (2012). Multi-neighborhood tabu search for the maximum weight clique problem. *Annals of Operations Research*, 196(1), 611–634. (Cited on pages 93, 104 and 106.)
- Zhao, X., Luh, P. B., & Wang, J. (1999). Surrogate gradient algorithm for lagrangian relaxation. *Journal of Optimization Theory and Applications*, 100(3), 699–712. (Cited on page 101.)
- Ziebuhr, M., & Kopfer, H. (2014). The integrated operational transportation planning problem with compulsory requests. In *International conference on computational logistics* (pp. 1–15). (Cited on pages 123 and 146.)

Ziebuhr, M., & Kopfer, H. (2016). Solving an integrated operational transportation planning problem with forwarding limitations. *Transportation Research Part E : Logistics and Transportation Review*, 87, 149–166. (Cited on pages [123](#) and [146](#).)

Yuan LI

Doctorat : Optimisation et Sûreté des Systèmes

Année 2017

Études sur la planification collaborative de transport entre transporteurs

Dans la collaboration entre transporteurs, plusieurs transporteurs forment une alliance pour échanger leurs demandes de transport dans le but d'améliorer la rentabilité. Dans cette thèse, nous avons étudié la planification collaborative de transport entre transporteurs de charges partielles. Plus concrètement, nous avons étudié trois sous-problèmes soulevés dans cette planification collaborative: le problème de ramassage et de livraison avec fenêtres de temps, profits et demandes réservées, le problème de détermination de gagnants dans l'échange combinatoire, et le problème de génération d'enchère.

Ces trois sous-problèmes sont les problèmes clés pour la planification collaborative de transport parmi des transporteurs, et ils sont peu étudiés dans la littérature. Nous avons établi les nouveaux modèles de programmation mathématique pour ces problèmes et développé des heuristiques efficaces pour trouver des solutions très proches de leurs optimums dans un temps de calcul raisonnable. Les heuristiques proposées sont plus performantes que les solveurs commerciaux (GUROBI, CPLEX) non seulement en termes de la qualité de solution, mais aussi en termes du temps de calcul.

Mots clés : logistique collaborative - optimisation combinatoire - programmation (mathématiques) - métaheuristiques - transport, planification.

Studies on Collaborative Transportation Planning among Carriers

In carrier collaboration, multiple carriers form an alliance to exchange their delivery requests for the purpose of improving profitability. In this thesis, we have studied the collaborative transportation planning (CTP) among less-than-truckload (LTL) carriers. More concretely, we have studied three sub-problems raised in this collaborative planning: the pickup and delivery problem with time windows, profits, and reserved requests (PDPTWPR), the winner determination problem (WDP) in carrier collaboration via combinatorial exchange (CE), and the bid generation problem (BGP).

These sub-problems are the key issues for collaborative transportation planning among carriers, and they are rarely studied in the literature. We have established new mathematical programming models for these problems and developed efficient heuristics to find solutions close to their optimums in a reasonable computational time. The heuristics proposed are more efficient than commercial solvers (GUROBI, CPLEX) not only in terms of solution quality, but also in terms of computation time.

Keywords: collaborative logistics - combinatorial optimization - programming (mathematics) - meta-heuristic - transportation, planning.

Thèse réalisée en partenariat entre :



Ecole Doctorale "Sciences et Technologies"