Università Degli Studi di Bergamo

Doctoral Thesis

Horizontal arc routing collaboration: models and algorithms

Author:
Dario Fontana

Tutor:
Prof. M. Grazia Speranza

External Supervisor:
Prof. Elena Fernandez

A thesis submitted for the degree
of Doctor of Philosophy
in Economics, Applied Mathematics and Operations Research

ABSTRACT

Nowadays, the interest in the class of collaborative transportation problems has been recently growing fast. However, the operational research literature about it is still quite scarce. Hence, In this work, we investigate horizontal collaboration, mainly on road transportation, among different companies, carriers and shippers. Background motivations for this thesis are: a still prevalent way of moving around goods in different parts of the world, an increasing pressure on logistic providers and an higher customer expectations about requested service. Therefore, we examine the existing literature on horizontal collaboration classifying different models and techniques used. An extensive and exhaustive literature review about these problems is given. We study profit, benefit and cost allocation procedures in order to better investigate the impact and the effectiveness of collaboration for the collecting of profits and the cutting of costs. Then, we develop a new profitable arc routing model to address a centralized partial cooperation among multiple carriers. We study two different formulations of this problem taking into account the impact of collaboration on the stand alone carrier profit. In the first one the goal is the maximization of the total profit of the coalition of carriers, independently of the individual profit of each carrier. The second variant includes a lower bound on the individual profit of each carrier. We formulate mixed integer programming models for the two variants of the problem and study their properties and their relations with well-known arc routing problems. We solve them with a branch-and-cut algorithm and quantify the impact of collaboration on a large set of instances. Finally, we develope two metaheuristic solution methods based on a Large Neighborhood Search and a Ruin and Repair heuristic framework. On one hand, we have a Variable Neighborhood Search (VNS), on the other hand, we have an Adaptive Large Neighborhood Search (ALNS). Both metaheuristics perform very well on a large set of instances solving almost all of them within few seconds. Moreover, both find feasible solutions on larger and more realistic instances within few minutes.

Ai Miei Genitori

Σημεῖόν ἐστιν, οὗ μέρος οὐθέν.

ACKNOWLEDGMENTS

This thesis arose out of research that has been carried out since I started my Ph.D. degree at the University of Bergamo and at the University of Brescia. By that time, I have worked with a great number of people whose contributions in assorted ways to the research and the making of the thesis deserve special mention. I would never have been able to finish my thesis without the guidance of those people. It is now a pleasure to convey my gratitude to them all in my humble acknowledgement.

First and foremost I want to thank my tutor M. Grazia Speranza. Thanks for believing in me and for constantly motivating and supporting my research with wise suggestions. It has been an honor to be one of her Ph.D. students. She pushed me to offer my best far beyond what I thought I was up to do. Moreover, she encouraged me to spend time for study and research in a foreign country, actually in Barcelona, Spain. This period of abroad study and research let me know a wonderful scientific environment at the UPC in Barcelona. In particular, I'm very grateful because I knew a great mathematician like Elena Fernandez. I have greatly benefited from her knowledge and experience, and I am thankful for her many advices. Moreover, she was very helpful and open minded and guided me through my stay in Barcelona and throughout my entire research.

I'm also thankful for the unique scientific environment available both at the University of Brescia and at the University of Bergamo and to their helpful respective staff. My friends and colleagues also played an important role. In particular I'd like to express my gratitude to Gianfranco, Michele, Claudia, Carlo, Enrico, Luca and Nicola. A big thanks goes to my fellows Ph.D. students Rosario, Marco, Sebastiano, Valentina and Diana who were always extremely supportive to me during these years. I also want to thank all the professors met during these years of university at the Catholic University of Brescia, at the University of Brescia, at the university of Bergamo and at the Polytechnic University of Catalonia.

I also acknowledge the support and the comradery of all my fellows and friends at the Convitto San Giorgio, in particular I wish to thank the head of it: Don Alessandro who always trusted and supported me. A special thanks goes to my friends Giovanni, Andrea, Alessio, Giuseppe, Christian, Roberto, Stefania and Meredith who shared with me some of the joy and the difficulties of this special time, helping me to enjoy my free time.

Last but not least, this list of acknowledgements would be incomplete without my deepest sincere appreciation for my parents, Giovanni and Iole, my uncle Giacomino, my brother Giorgio, my sister Claudia, her husband Matteo and her daughter Emilia. Only with their support when hard times arrived I was able to continue this journey.

Finally, I would like to thank everybody who was important to the successful realisation of this thesis, as well as expressing my apologies to those I could not mention personally, I must share with these individuals any merit found in this research. Any shortcomings remain, of course, my own.

CONTENTS

Abstract ii

Acknowledgments iv

Contents vi

List of Figures ix

List of Tables x

List of Algorithms xii

1. Introduction 1

1.1 Structure of the thesis 2

2. Survey on Routing Collaboration 4

2.1 Introduction 4

2.2 Horizontal Collaboration 6

2.3 Literature review 8

2.4 Profit sharing 12

2.5 Conclusions 19

3. On the Collaboration Uncapacitated Arc Routing Problem 21

3.1 Introduction 21

3.2 Literature review 24

3.3 Collaboration Uncapacitated Arc Routing Problem 26

3.4 Theoretical results 35

3.5 The branch-and-cut algorithm 40

3.6 Data generation and computational results 41

3.7 Conclusions 50

4. Metaheuristics for the Collaboration Uncapacitated Arc Routing Problem 51

4.1 Introduction 51

4.2 The destroy and repair heuristic framework 52

4.3 The initial solution 54

4.4 Neighborhoods 58

4.5 The destroy and repair cycle 58

4.6 Experimental Analysis 65

4.7 Conclusion 72

5. Conclusions 73

Bibliography 76

Bibliography 76

Appendix 84

A. Electronic Appendix 85

LIST OF FIGURES

3.1 Example 33

LIST OF TABLES

3.1 Summary of Example 3.3.1 34

3.2 Summary of Example 3.3.1 35

3.3 Instance summary 43

3.4 Number of instances for each combination of parameters . . . 44

3.5 Results for CUARP 46

3.6 Results for t-CUARP, $\beta = 0.5$ 47

3.7 Results for t-CUARP, $\beta = 1$ 47

3.8 Instance summary after moving depots 48

3.9 Results for CUARP after moving depots 48

3.10 Instance summary with 3 depots 49

3.11 Results for CUARP with 3 depots 49

3.12 CUARP G16 results with increasing size of shared sets 49

4.1 Large instances characteristics 66

4.2 VNS results for CUARP instances 67

4.3 ALNS results for CUARP instances 68

4.4 relaxed CUARP results 69

4.5 Metaheuristics results on large instances 70

LIST OF ALGORITHMS

4.1 Initial Solution Procedure 54

4.2 Stopping Criterion 59

4.3 Variable Neighborhood Search 61

4.4 Adaptive Large Neighborhood Search 63

1. INTRODUCTION

Within this thesis, we investigate the horizontal collaboration, mainly in road transportation, among different companies, carriers and shippers. Indeed, nowadays transportation companies acknowledge growing pressure and road transportation is still a prevalent way of moving around goods in various parts of the world. Horizontal collaboration means cooperation among transportation firms and companies that operate at the same level of the supply chain performing comparable activities. The interest in horizontal collaboration on road transportation among different players in routing problems has been recently boosted by higher pressure on logistic providers and by higher customers expectations about requested services. However, in this field literature is still quite scarce. We give an extensive literature review, covering also impediments, driving forces and opportunities linked to it.

We develop a new arc routing model to address the problem of carrier collaboration in a centralized framework. We call it the Collaboration Uncapacitated Arc Routing Problem (CUARP), an uncapacitated arc routing problem with multiple depots, where carriers collaborate to improve the profit gained. We also propose variants that take into account profit thresholds. We propose a large set of benchmark instances and solve them with exact and metaheuristic methods taking into account and underlying the impact of collaboration for collecting profits and cutting costs. Finally, by means of metaheuristic methods we are able to find feasible solutions for larger and more realistic instances.

1.1 Structure of the thesis

The thesis is organized as follows.

- **Chapter 2: Survey on Routing Collaboration.**

  In this chapter we survey main contributions appeared in the literature on Routing Collaboration Problems. We focus on models and solving methods. This chapter provides also a survey of the major game theory profit allocation techniques used in Routing Collaboration Problems.

- **Chapter 3: On the Collaboration Uncapacitated Arc Routing Problem.**

  In this chapter a new arc routing problem for the optimization of a collaboration scheme among carriers is presented. We focus on situations where collaboration is managed in a centralized way. We consider a set of carriers cooperating under the guidance of a central station that acts in a non-partisan way. This yields to the study the Collaboration Uncapacitated Arc Routing Problem (CUARP), an uncapacitated arc routing problem with multiple depots, where carriers collaborate to improve the profit gained. We study two variants of the CUARP, solve the formulations for the two proposed variants with a branch-and-cut algorithm, and quantify the impact of collaboration for a large set of benchmark instances.

- **Chapter 4: Heuristics for the Collaboration Uncapacitated Arc Routing Problem.**

  In this chapter heuristic approaches based on Large Neighborhood Search are developed in order to solve CUARP instances of realistic size. We propose two different heuristic frameworks a Variable Neighborhood Search (VNS) and an Adaptive Large Neighborhood Search (ALNS). We solve a set of CUARP benchmark instances to prove the effectiveness of the proposed algorithms. Finally, we use the VNS and the ALNS to find feasible solutions for larger and more realistic instances.

- **Chapter 5: Conclusions.**

  This chapter contains the concluding remarks of the thesis and future research prospectives and open areas to deal with.

- **Electronic Appendix.**

  This appendix has an attached .xlsx file containing the characteristics of all instances and the results of all experimentations done within this thesis.

2. SURVEY ON ROUTING COLLABORATION

2.1 Introduction

Road transportation is still a prevalent way of moving around goods in various parts of the world. The interest in the class of collaborative transportation problems has been recently growing fast. Indeed, collaboration is widely seen as one of the best ways to deal with increasingly complex business sectors in order to create an advantage, as pointed out in Fugate et al. [43], Stefansson [80] and Cruijssen [26]. Generally, collaboration among companies may indicate different levels and kinds of cooperation. Indeed, companies may collaborate on sharing information about customers and core or non-core business activities, such as commodity purchasing or distribution. In particular, we can have collaboration among companies in goods procurement and distribution at different levels of the supply chain. Therefore, we can distinguish between vertical and horizontal collaboration.

Vertical collaboration is characterized by interactions among different levels of the supply chain, and shippers, carriers and customers cooperate to improve services quality. Among the most known examples of vertical collaboration there are the Vendor Managed Inventory (VMI), the Efficient Customer Response (ECR) and the Collaborative, Planning, Forecasting and Replenishment (CPFR). Moreover, transportation companies may vertically collaborate setting up distribution systems with inter-modal exchange nodes to allow collaboration among different levels of the supply chain. For further studies and comprehensive surveys on vertical collaboration see Fawcett et al. [39], Fugate et al. [43], Kilger et al. [52] , Thomas et al. [81] and Simatupang et al. [79]. Wallender et al. [84] and Dong et al. [33] give an overview of VMI, while Holström et al [50], Seifert [74] and Sherman [78] give a comprehensive analysis of ECR and CPFR.

On the other hand, we have *horizontal* collaboration when companies at same level of the supply chain cooperate, even if they are competing companies. Generally, we can distinguish four kinds of horizontal collaboration, as pointed out in [26]. Firstly, we have *competition* when companies move on an action-reaction arrangement to reach comparable goals. Secondly, companies *coexist* if they have relationships which do not include common objectives, money flows and collaboration on business core activities. Then, *cooperation* is characterized by developing tight bonds among companies, in order to reach a common goal. Finally, if there is a collaboration agreement that obliges companies to cooperate to reach a common objective, but outside this agreement there is strong competition, we have *co-opetion*. This last model of horizontal collaboration is the most widespread because it allows companies to collaborate in some, but not necessarily all, activities, within a framework of fixed rules. For our purposes in this chapter, we defined *horizontal collaboration* as *cooperation between two or more firms that are active at the same level of the supply chain and perform a comparable logistics function on the landside*, as proposed by Cruijssen et al. [26]. Horizontal collaboration is very well known in ocean shipping and air transport literature. For comprehensive surveys on maritime collaborative transportation we address the reader to Clarke [22] and Sheppard et al. [77], while for airline industry collaboration we suggest Fan et al. [38], Oum et al. [64] and Park [68]. On the contrary, literature in this field is quite scarce. Few papers with relevant models have been published and several problems on collaboration routing remain to be studied. We concentrate our attention on classifying different models; a recent survey giving a broad literature review can be found in Verdonck et al. [83]. Nevertheless, we cover briefly the impediments, the opportunities and the driving forces and events that lead companies and more generally road transportation logistic providers to collaborate.

Background and motivations for the survey

Nowadays, road transportation companies have tools that allow them to decrease the costs of serving customer orders. These tools, however, are often limited by the growing competitiveness among companies. This drawback and severe limitation can be prevented by means of collaborative schemes. Indeed, transportation providers that participate or form coalitions in order to fulfill customer requests may reduce their costs, for instance balancing their customer sets and/or reinforcing their market position. Hence, a coalition yields global and individual benefits to its partners.

In this chapter we present a survey of the literature on horizontal collaboration routing problems on road transportation. Section 2.2 gives a broad introduction to horizontal collaboration and its driving forces, impediments and opportunities. A literature review is given in Section 2.3. In Section 2.4 we review the profit allocation schemes and their application on horizontal collaboration routing problem. Conclusions are drawn in Section 2.5.

2.2 Horizontal Collaboration

In road transportation horizontal collaboration is the collection of concerted practices and strategies among transportation companies, which operate at the same level of the supply chain, to increase their performances, as pointed out in Cruijssen et al. [27]. In this section we revise the driving forces, the impediments and the opportunities of horizontal collaboration and propose a different classification to deal with such problems. Indeed, the literature has grown fast during last years, but it lacks a formal classification.

The main driving force for horizontal collaboration is each companies selfish expectation of a positive benefit. Hence, transportation companies form alliances to collaborate and to jointly achieve a common goal. These collaborations develop because of synergies among companies. Among the most known synergies we can distinguish between economies of scale and economies of scope. Economies of scale indicate the decrease in unitary costs by producing more of the same commodity or providing a particular service more frequently and/or to more customers. In Cruijssen et al. [25] such economies are exploited in a joint route planning on horizontal roadside collaborative transportation. Economies of scope refer to cost savings or profits because of the addition of new products or new services. Usually, such economies drive companies to collaborate among them. Specifically, in road transportation economies of scope enable logistic providers to offer services and goods not available outside the collaborative network and framework. Moreover, collaboration may allow companies to enlarge their orders and increase their sets of customers. Hence, another driver is the will and the opportunity to strengthen each companies market position. Furthermore, horizontal collaboration among roadside transportation companies lead them to cut costs by making more profits and savings.

Horizontal collaboration has not only to be seen from a cost-reduction perspective, tough this is by far the most important driving force for companies. For instance, a side effect that has to be taken into account is the reduction in polluting emissions (*e.g.* Ballot et al. in [14] and Pan et al in [67]) because it links cost savings to an increasing demand of reducing emissions.

Nevertheless, many coalitions, alliances, collaborative networks and agreements fail, even if academic works and studies are more focused on successful cases and theoretical analysis. Major difficulties can be found during the implementation of a collaborative agreement that binds each transportation company to behave itself according to rules and fixed frameworks. Indeed, to set up a clear and truthful collaboration, trust among partners becomes very important. For instance, some companies can act in an opportunistic way to improve their position and gain more. Hence, truthfulness is highly requested during information sharing for making a collaborative network work properly. Recent developments in information and communication technology help and make collaboration among different partners more easy. However, a key point that usually leads to collaboration failure deals with profit sharing. Indeed, some companies may be reluctant to collaborate because they believe their profit share to be unfair. Anyhow, the allocation of profits or costs may also be used as a driver to success for collaborative networks, if it is done in a right way so as not to annoy any partners. Hence, another key point in horizontal collaboration is how to form coalition such as to ensure long-term functioning networks. Notably, cooperative game theory (CGT) tools (see Section 2.4) are very useful to deal with this kind of issues, in order to take into account the distribution of power, the level of geographical and business synergy, the willingness to cooperate, and exchange assets or customers.

In particular, we can note two different versions of horizontal roadside collaboration: centralized and decentralized. Ideally, the existence of a central third party, able to gather all informations in a truthful way is probably the best choice to boost up companies profits. Indeed, it may react quickly and in a clever way to sudden changes and adapt its behavior to dynamic instances. Moreover, an external player that knows all about customers, orders and logistic assets of companies can solve large instances optimally assigning, clustering and handling demands and companies efficiency and production capacities. However, this can lead to a merge of the companies and/or an assimilation of small companies into the biggest ones. We, intentionally, avoid to deal with this issue because it is far beyond our purposes. Nevertheless, it is clear that even in a decentralized approach trust among players that participate to the collaborative scheme and agreement is a major point. Indeed, some companies may lie or behave in an opportunistic way in order to achieve a dominant position and then gaining more from the collaboration. This point can be addressed in various ways, enforcing controls, establishing strict and binding rules or reducing the information exchange to a minimum.

To sum up, cost savings, profit increasing and logistic benefits are the most important drivers for roadside collaboration. There are also some opportunities, other than cost savings, like polluting emissions reduction and merging possibilities among companies. However, we have also impediments such as opportunistic behaviors and unfair benefits allocation among the most common ones.

2.3 Literature review

Analysing the existing operational research literature on horizontal roadside collaboration, we notice that a major part of the articles are devoted to study carrier alliances and cooperation in which customers and orders are shared or exchanged. In these papers logistic assets such as depots and vehicles fleets are left unchanged. On the other hand, some papers deal with carrier collaboration sharing vehicle capacity, depots and other logistic assets. In [30] Dai and Chen approach logistic collaboration in less-than-truckload carrier networks mixing order and vehicle capacities and fleet sharing. They suggested a mathematical formulation within a context of centralized horizontal collaboration in which a third party logistic provider receives all the orders and then distributes them among carriers sharing vehicle capacities.

Hence, we review the horizontal roadside collaboration dividing it into two streams.

2.3.1 Customers sharing

Current research focuses on different techniques to optimally deal with the re-allocation of orders, customers and services. A majority of the papers deal with joint-route planning or auction based mechanisms. However, approaches like bilateral lane exchanges, load swapping and other dispatching policies are investigated.

Joint-Route planning

A basic and widespread approach to customers sharing is joint route planning. We describe as a joint-route planning technique the procedure that combines all carriers and all customers orders in a central logistic provider station with the aim to produce efficient routes for all carriers and for the whole network of carries. This approach leads to economies of scale, reducing travel distances, costs and empty truckload movements and merging regions of distribution (*e.g.* Crujissen et al. in [25]). Crujissen et al. in [25] develop a framework based on Vehicle Routing Problem (VRP) with time windows (VRPTW) in order to investigate the synergy of horizontal collaboration under a joint-route approach. In this paper the objective is the minimization of the routes lengths taking into account that each of them starts and ends at an origin node and that customers demand does not have to exceed vehicles capacity. A heuristic is developed based on the original savings heuristic by Clarke and Wright [21] and on more recent developments like those described in Liu and Shen [59]. Another variant of the VRP, such as the multi-depot pick-up-and-delivery problem with time windows, is used to model a collaborative carrier customer sharing problem by Krajewska et al. [54]. The problem consists in finding a feasible set of route to minimize costs under time windows constraints (MDPDPTW). This model is tested on real life instances and data provided by a German freight forwarder. A large neighborhood search is developed to solve it based on Ropke and Pisinger [72] and Shaw [76] heuristics. Computational results show that participants in coalition and collaboration may achieve significant cost savings and benefits. In Dahl and Derigs [28] MDPDPTW is used to address an order sharing problem in a collaborative network of independent carriers in a dynamic perspective. This problem is solved modifying the ROUTER indirect search heuristic previously developed by Derigs et al. in [32]. Simulations and computational results on a large set of real data provided by 50 European express carriers show that costs may be reduced up to 13%. Liu et al. in [58] formulate the joint-route planning problem as a multi-depot capacitated arc routing problem with full truckloads (MDCARPFTLs). To solve it they adjust and extend a greedy algorithm described by Ergun et al. in [36]. Results show robust and high quality solutions reached in reasonable computational times. Instead of considering the whole carrier network for collaboration, in Bailey et al. [13] possible reductions in empty backhauls by adding customers and orders from collaborative partners are investigated. Two different mixed-integer programming (MIP) models are developed and solved through basic tabu search approaches. Computational experiments show that backhaul cost savings may reach up to 28%.

Auction-based mechanisms

Order and customer sharing is also carried out through auction based mechanisms. Firstly, each carrier defines customers to be shared in a cost efficient manner using various optimization methods. Then, orders and customers are shared employing profit auction mechanisms. It is worth noting that auction-based mechanisms leads directly to a profit allocation since carriers reward their partners during the auction; while in the joint-route planning schemes a fair benefit allocation procedure has to be built up.

Figliozzi in [40] proposes an auction based mechanism making use of reservation prices to reassign customers in a dynamic environment. In particular, the paper focuses on an incentive compatible collaborative mechanism. Instead, Krajewska et al. in [53] describe a request allocation procedure which relies on combinatorial auctions and cooperative game theory tools to optimize collaborative profit and shared profit, respectively. Combinatorial auctions are also used in Berger et al. in [17]. In this paper, however, the authors focus on the amount of information shared among partners in a coalition. Computational experiments and simulations show that a decentralized approach can be useful if carriers are not willing to share a fair amount of information. Clearly, if informations are correctly shared they demonstrate that a centralized approach is superior. Hence, they underline that collaborative carrier profit increases in line with the increasing of information sharing. Dai and Chen in [29] split the auction-based mechanism in two decision problems. Firstly, carriers select customers they do not want to serve themselves and then they identify desirable customers or a subset of customers from those shared by other carriers. The authors apply this model to a less-than-truckload (LTL) problem allowing multiple auction processes to happen simultaneously. This approach led to a great level of interaction among carriers. Moreover, simulations on 20 randomly generated instances reveal that profit gained through collaboration is significantly greater than the stand-alone profit for each carrier. Contrary to these studies and all auction-based sharing mechanisms, Wang and Kopfer in [85] propose a combinatorial auction method in which carriers exchange complete vehicle routes and not only single customer services. The underline problem is a pick-up-and-delivery with time windows. This problem is solved within 3 stages. First, carriers offer all their customers and each carrier determines appropriate prices for route transfers. Then, the last two stages consist of a bidding generation process and the winner determination of the auction, respectively. Wang and Kopfer state that their collaborative scheme achieves cost reductions between 2% and 18%, by means of computational experiments on instances generated for this purpose.

Other policies to implement customers sharing are the bilateral exchange mechanism investigated by Özener et al. in [66], the load swapping suggested by Clifton et al. in[20] and time and quantity strategic dispatching policies investigated by Zhou et al. in [86]

2.3.2 Capacity sharing

As pointed out previously, carriers may also collaborate by sharing vehicle capacities. In this subsection we provide a general overview of the various techniques, solution approaches and application areas of papers dealing with collaboration through vehicle capacities sharing.

Argawal et al. in [1] study the problem of sharing the capacities in liner shipping industries. Indeed, in liner shipping industries carriers often cooperate pooling together their fleet and their capacities. They formulate a MIP model and test three different heuristics to solve it comparing their results. These heuristics are a greedy algorithm, a column-generation based algorithm and a Benders decomposition-based algorithm. Computational experiments carried out show that significant improvements may be achieved through collaboration. In road transportation capacity sharing is studied by Herndández et al. in [48]. They investigate a less than truckload dynamic carrier collaboration problem, modeling it as a MIP. Capacities are time dependent and the collaboration is managed in a centralized way to minimize costs. Instead on focusing on cutting costs, Houghtalen et al. in [51] maximize collaborative profit formulating a MIP to model capacity sharing. Two variants of the model are proposed. On one hand a so-called Limited Control model restricts carriers individual decisions, while, on the other hand, a so-called Strict Control model allows carrier a greater freedom and greater control over the whole collaboration process. A comparison of these variants leads to demonstrate that allowing greater overall control to each carrier causes a decrease in profit collecting.

2.4 Profit sharing

In this section we review profit allocation schemes and their applications in some existing papers. Let us consider a logistic framework in which multiple landside transportation companies operate. As pointed out in previous sections companies have multiple reasons to collaborate to enhance their services and increase their profits. Hence, collaboration schemes lead transportation providers to form coalitions in order to ensure such improvements. Coalition formation generates the problem of how to share the whole benefits gained among companies. Therefore, in this section, we deal with profit sharing problems arising in a collaborative transportation framework.

Proportional allocation

Simple rules usually arise in case of cooperation in order to determine and divide benefits. Commonly, transportation companies reach an agreement to share costs, savings, gains and profits proportionally to a single indicator. Let ω_i be the weighted value for company i. Then, companies may split the total profit gained proportionally to their weighted values. We can compute ω_i in different ways:

- proportional to the number of customer orders collected before collaboration;
- proportional to the profits or costs before collaboration;
- proportional to the quantity of informations owned before collaboration;
- proportional to the quantity of logistics owned before collaboration.
- proportional to the number of customers actually served;
- proportional to the quantity of total load actually delivered;
- proportional to the distance actually travelled;

- proportional to the number of vehicles actually employed.

We note that these rules are of two kinds: with the first four rules, companies share profit proportionally to a value decided before collaboration, while the last four rules are proportional to a value decided after collaboration has taken place.

Cooperative game theory

At a first approach, proportional allocation may be regarded as fair, however in the long run some companies may be unsatisfied because their true or their actual contributions to the coalition is disregarded and undervalued by its partners. Hence, to deal with these drawbacks, we introduce cooperative game theory and some of its basic concepts, definitions and properties. For a comprehensive survey on cooperative game theory we refer to Driessen [34] and to Tijs et al. [82].

A *cooperative game* consists of a finite number n of players and a characteristic function. Players can form different *coalitions* in order to achieve a better result in the game, we define a coalition as a subset S of the set of all players. The coalition N of all players is called grand coalition. The characteristic function $v : 2^{\mathbf{N}} \to \mathbb{R}$ is a function from the set of all possible coalitions of players to a set of payments such that:

$$v(\emptyset) = 0;$$
$$v(S \cup T) \geq v(S) + v(T), \quad \forall S, T \subseteq N, S \cap T = \emptyset.$$

Each coalition is associated with a payment and each player has its own share of payment. Hence, the characteristic function ensures us that an empty coalition has a zero value and that two coalitions after merging have at least the same value as playing separately. Then, we can interpret the numerical value associated with each coalition, by means of the characteristic function, as a measure of its stand alone profit or its contribution to the coalition N. Moreover, $v(N)$ is the value associated with the grand coalition and the following inequality holds

$$v(N) \geq v(S) \quad \forall S \subset N$$

We denote by $p(S)$ total profit generated by S and p_j as profit allocated to j player. Below, we introduce some basic desirable properties.

1. *Efficiency/Budget balance*
   $\sum_{j \in N} p_j = v(N)$,
   an allocation is efficient if the total profit is split among all players. Then property is also called budget balance property;

2. *Individual rationality*
   $p_j \geq v(\{j\})$,
   an allocation is said to be individually rational if no player can gain less than is stand alone profit;

3. *Symmetry*
   if $v(\{j\}) = v(\{i\})$ then $p_j = p_i$,
   an allocation is symmetric if it allocates equal shares of profit to players with equal stand alone profits;

4. *Dummy player*
   if $v(S) - v(S \setminus \{j\}) = v(\{j\})$ for each S and $j \notin S$ then $p_j = v(\{j\})$,
   a dummy player, which gives marginal contribution to any coalition, must gain exactly its stand-alone profit;

5. *Additivity*
   $p(S \cup T) = p(S) + p(T)$ if $S, T \subset N$,
   which means that the allocation to a player in a sum of two games is the sum of the allocations to the player in each individual game.

6. *Stability*
   $\sum_{j \in S} p_j \geq v(S)$, with $S \subseteq N$,
   a stable allocation provides that none of the players or coalition of players could decide to opt out and form new coalitions to achieve greater profits.

A key concept in cooperative game theory is the *core* of a game which is defined as the set of all stable allocations. The core is formed by those payments allocations $p_1, \dots p_n$ that satisfy the following conditions

$$\sum_{j \in S} p_j \geq V(S), \text{ with } S \subset N$$
$$\sum_{j \in N} p_j = V(N).$$

The former condition, which is the stability Property 6 prevents players from colluding to form a subcoalition in order to gain more. In the case $S = \{j\}$ it ensures that each player receives at least what he could get on his own and it is equal to individual rationality Property 2. The latter condition implies that the allocations $p_1, \dots p_n$ split the total value gained by the grand coalition. This condition is the efficiency/budget balance Property 1. Given the above two conditions, no player has an advantage by leaving the grand coalition and the profit allocation is called *stable*.

On the basis of these properties, there are many different payments allocations suitable as solutions for a cooperative game. Firstly, we introduce for each coalition S and for each profit allocation P the *imputation*

$$I(P,S) = \sum_{i \in S} p_i - v(S), \forall S \subseteq N,$$

which measures how far an allocation is from the core, by computing the difference between the sum of profit allocated to single players and the profit gained by the coalition of those players. Next, we present the most useful and widespread methods to allocate profit among players in a coalition.

The Shapley Value

A well known allocation method is the Shapley value introduced by Shapley in [75] and defined as:

$$p_j = \sum_{S \subset N} \frac{(|S|-1)!(|N|-|S|)!}{|N|!}[v(S) - v(S \setminus \{j\})], j \in S. \tag{2.1}$$

The Shapley value is the average weighted marginal contribution to the coalition of all transportation companies involved in the grand coalition. The Shapley value is known to be efficient, symmetric, dummy and additive. However, the Shapley value does not always belong to the core, this means that the allocation provided by it is not always stable.

The nucleolus

Another well known allocation method is the nucleolus. The nucleolus was introduced by Schmeidler in [73]. It is an allocation method that lexicographically minimizes the maximal excess.

$$
\begin{aligned}
&\min \varepsilon \\
&\quad I(P,S) = \sum_{i \in S} p_i - v(S) \leq \varepsilon \qquad S \subset N, S \neq \emptyset \\
&\quad \sum_{i \in N} p_i = V(N) \\
&\quad p_i \geq v(\{i\}) \qquad i \in N
\end{aligned}
$$

The nucleolus always exists, and is efficient, individually rational, dummy and symmetric. Moreover, it fulfills the stability property whenever the core is non empty.

τ-Value

An allocation method proposed by Tijs et al. [82] is the τ-Value. The τ-Value for each player i is a linear combination between the lower and the highest value it can achieve. The highest achievable value is given by $M_i = v(N) - v(N \setminus \{i\})$ and is also called the utopia payoff of player i. While, the lower value is given by the maximum over all possible coalitions of what can remain to player i if all other players gain the utopia payoff. Among all possible linear combinations those which fulfill the efficiency property are chosen. The τ value for player i is therefore defined as:

$$
\begin{aligned}
p_i = \tau_i := m_i + \alpha(M_i - m_i)& \\
M_i &= v(N) - v(N \setminus \{i\}) \\
m_i &= \max_{S:i \in S} R(S,i) \\
R(S,i) &= v(S) - \sum_{i \in S \setminus \{i\}} M_i \\
\sum_{i \in N} \tau_i &= V(N)
\end{aligned}
$$

The τ-Value is not stable, since it does not belong to the core. However, efficiency, individual rationality, dummy player and symmetry properties hold. This method is used among other ones in Lozano et al. [60].

Gap function method

The Gap function method described in [82] can be seen as a variant of the τ-Value. Differently from the τ-Value it minimizes the difference between the sum of the utopia payoffs of coalitions and what coalitions actually gain:

$$p_i = \lambda_i := \min_{S:i\in S} g(S)$$
$$g(S) = \sum_{i\in S} M_i - v(S)$$
$$M_i = v(N) - v(N \setminus \{i\})$$

This method has all the properties of the τ-Value except for efficiency. Indeed, it may not be efficient because it does not ensure that all profit gained is redistributed among players. Frisk et al. in [42] have used this method to allocate costs in a collaborative forest transportation environment.

Next we list some methods developed in Tijs et al. [82] to allocate cost and profit among different players in a cooperative game theoretical perspective.

Equal Profit Method

Let us consider the relative savings of participant i as $\frac{v(\{i\})-y_i}{v(\{i\})}$. The *Equal profit method* is a stable allocation that minimizes the maximum difference in pairwise relative savings. Hence we need to solve the following LP problem:

$$\min \epsilon$$
$$\frac{p_i}{v(\{i\})} - \frac{p_j}{v(\{j\})} \leq \epsilon \qquad i,j \in N$$
$$\sum_{i\in S} p_i \geq V(S), \qquad S \subset N$$
$$\sum_{j\in N} p_j = V(N)$$

Other allocation methods, described in the literature, are the following. They are all based on proportional allocation of Marginal Cost $MC = v(N) - \sum_{j\in N}[v(N) - V(N\setminus\{j\})]$, which can be seen as the difference between the profit of the grand coalition and the marginal profit gained by player i.

Equal Charge Method

The Equal charge method allocates proportionally MC among all players.

$$p_j = \frac{MC}{|N|}$$

Alternative Cost Avoided Method

This method allocates profit weighted them using the cost avoided by each player.

$$p_j = \frac{MC}{v(\{j\}) + V(N \setminus \{j\}) - v(N)}$$

Cost Gap Method

The Cost Gap Method weights the MC with ω_i for each player i.

$$p_j = \frac{MC}{\omega_j}$$

such that

$$\omega_j = \min_{S:j \subset S} \gamma(S)$$
$$\gamma(S) = v(S) - \sum_{j \in S} [v(S) - V(S \setminus \{j\})]$$

Hence, ω_i can be viewed as the minimum difference between profit $v(S)$ and marginal profit $V(S \setminus \{i\})$ among all coalitions S containing player i.

Next, we briefly review the use of game theory in roadside collaborative transportation. In Krajewska et al. [54] the potential of modern transportation systems is studied with a unique combination of routing, scheduling and game theory approaches. In particular, they analyzed request allocation and profit sharing in an horizontal collaboration framework. Moreover, they discussed various ways of fairly sharing profits giving numerical results for real and artificial instances. In Lozano et al. [60] horizontal collaboration is addressed as a way to reduce costs. They use an optimization model to solve different collaboration scenarios thus testing which one is the most profitable. Moreover, the allocation problem is tackled by means of cooperative game theory comparing different game solution concepts, notably the Shapley value and the Nucleolus. Dai et al. in [31] introduced a new optimization model for carrier collaboration in pickup and delivery service proposing three ways of allocating profits based on Shapley value. Ozener et al. in [65] studied a procurement collaborative transportation network developing cost-allocation methods based on well known cooperative game theory properties. Then, they performed computational studies on random and real life data to test allocation schemes performances. Audy et al. in [9], [11], [10], [12] investigated the impact of benefit sharing in various real life applications, from the furniture industry to freight collaboration, using game theory tools and testing different allocation scheme. Comparisons among various schemes and procedures to divide costs and profits among partners are carried out by Berger et al. in [17] and by Liu et al in [57].

2.5 Conclusions

In this chapter we have provided a comprehensive overview of horizontal roadside transport collaboration. We classified existing works into various streams which sometimes overlap. Indeed, firstly we consider centralized and decentralized approaches, then we analyze the literature on the basis of what is shared: customer/orders or logistic assets. Finally, we evaluate the use of cooperative game theory tools to solve the related problem of allocating cost and/or profits, focusing on methods that go beyond a mere proportional allocation. Nevertheless, even if this kind of problems attracts the interest of the OR community, the number of papers dealing with this class of problems is very limited.

Potential future research opportunities include the developments of new models to address realistic problems and the drawing up of more general and rich mathematical formulations to take into account various aspects of horizontal roadside collaboration. Since the main goal of different participants in a collaborative scheme is making profit, it can be interesting to refine old and develop new efficient profit or cost allocation procedures through the use of cooperative game theory tools and then the matching of these techniques with particular problems. Finally, it may be stimulating to pursue an integration of more levels of supply chain in a collaborative perspective.

3. ON THE COLLABORATION UNCAPACITATED ARC ROUTING PROBLEM

This chapter is entirely based on the paper "On the Collaboration Uncapacitated Arc Routing Problem" coauthored with Elena Fernandez and M. Grazia Speranza. This paper has been accepted for publication in "Computers and Operations Research".

3.1 Introduction

Collaboration among carriers becomes more and more valuable because of surging pressures to improve profitability and to reduce costs. Nowadays, collaborative transportation is regarded as one of the major trends in transportation research. Indeed, increasing carrier insurance and fuel costs combined with a more intense market competition lead carriers to look for new and more efficient solutions. Primarily, carriers focus on reducing costs looking for efficient route planning and scheduling. These costs are strongly correlated with the location of customers. Whereas a carrier would benefit from having its customers concentrated in the same area, for a number of reasons they may end up being geographically dispersed. This forces the carrier to create long routes for its vehicles, with associated high cost in terms of vehicles usage and drivers time. It is often the case that customers that are inconveniently located for a carrier are conveniently located for a different carrier. Thus, a collaborating set of carriers can redistribute the customers, opening up, through collaboration, cost saving opportunities otherwise non achievable.

In general, there are different types of carriers: general, regional or functional. The general carrier is non specialized and has the assets and the logistics to serve all its customers taking care of all kinds of item distributions. Instead, a regional carrier is more bound to a defined geographical service area whereas a functional carrier serves a specific market or specific goods that require a specialization in transportation. Hence, for instance, a regional carrier can rely on a general one to serve customers outside its service area, or a general carrier can choose to handle particular goods (such as furniture, frozen foods) through a functional carrier.

Logistic collaboration can be pushed further considering that it allows carriers to increase the average load of the vehicles. In fact, also in the case the customers are located in the same area, the load to be delivered in a trip by a carrier may be substantially lower than the vehicle capacity and make the individual trip non profitable. A carrier that has to deliver a certain amount of goods that fills only part of the capacity of its smallest vehicle may borrow a vehicle of the right size from another carrier or transfer the load on a vehicle of another carrier traveling to the same area at the same time.

Increasing attention to the environmental impact of emissions in cities represents an additional strong motivation to study collaboration among carriers, since local authorities increasingly push carriers to find new policies and new technological and logistical solutions that improve city logistics. In [71] challenges and pressures faced by carriers to cooperate to make urban freight transport more efficient are pointed out, and best practices actually brought into practice in The Netherlands are presented.

Recently, collaboration has been enhanced by advances in information and communication technology that have enabled information sharing among carriers. Information can be shared in two alternative ways. In a centralized collaboration scheme, a central decision maker redistributes customers and/or logistic assets among carriers. This decision maker may be a third party who acts in a non-partisan way or may be a large carrier that resorts to other carriers to manage all its orders and customers. In a decentralized collaboration scheme, carriers exchange their orders individually or in clusters. In this case, carriers cooperate at the same level trusting each other for the information shared. All the above considerations and approaches apply to both truckload or less-than-truckload carriers.

In this chapter, we focus on situations where collaboration is managed in a centralized way. We consider a set of carriers cooperating under the guidance of a central station that acts in a non-partisan way. Each carrier has a depot and a set of customers. Each customer is represented with an arc and its service generates a revenue. Each carrier identifies a subset of customers that it wants or needs to serve. These customers may be the most easily served, the most profitable or the most strategic ones. The remaining customers are defined as shared customers, that is customers that may be served by other carriers. A shared customer may end up being served by the carrier that decided to share it, when combined with customers shared by other carriers. Part of the revenue of a shared customer goes to the carrier that decided to share the customer and part goes to the carrier that actually serves it. We allow a shared customer not to be served by any carrier of the coalition. In this case the revenue is not collected by any carrier. This corresponds to the situation where the customer is not profitable for any carrier of the collaborating group and in a further phase a different and interested carrier will be searched. We assume that each carrier has one vehicle and that vehicle capacity is not relevant, that is the vehicles are uncapacitated.

The motivation for studying this problem comes from potential applications. In general, applications arise in private companies offering services which allow competition and collaboration, and where customers may be modelled as arcs of a network. As an example we mention home pick-up and delivery, including private mail and small packaging distribution, and taxi services. For example, the problem that we address can model a group of independent taxi drivers collaborating under the guidance of a central station.

We call the proposed problem, that may be seen as belonging to the class of arc routing problems with profits, Collaboration Uncapacitated Arc Routing Problem (CUARP). We study two different variants of the CUARP. In the first one the goal is the maximization of the total profit of the coalition of carriers, independently of the individual profit of each carrier. The second variant includes a lower bound on the individual profit of each carrier. This lower bound may represent the profit of the carrier in the case no collaboration is implemented. We formulate mixed integer programming models for the two variants of the problem and study their relations with well-known arc routing problems. We also look at the CUARP from a game theory perspective. As it is usual in arc routing problems, the proposed formulations have a number of connectivity constraints which is exponential in the number of customers. This leads us to study the separation problem for such constraints. We solve the formulations for the two proposed variants with a branch-and-cut algorithm and quantify the impact of collaboration. Starting from 118 benchmark instances for the Privatized Rural Postman problem, we generate a total of 971 instances, with 2 or 3 carriers and varying characteristics, such as different locations of the depots and different thresholds for the profit. We solve all instances within few seconds. On each instance we compare the optimal solution obtained in the case where no collaboration is allowed with the case where collaboration is allowed, and show that the profit of the coalition increases up to twice or even three times the profit achieved without collaboration.

The rest of the chapter is organized as follows. Section 3.2 introduces the relevant literature. The two variants of the CUARP are formally described and formulated in Section 3.3. Section 3.4 presents the theoretical results. In Section 3.5 we describe the separation procedure for the connectivity constraints that is used in the branch-and-cut algorithm. Data generation and computational experiments are described in Section 3.6. Finally, conclusions and future work are discussed in Section 3.7.

3.2 Literature review

The literature on collaboration in transportation can be divided in two streams, one on vertical and the other on horizontal collaboration. Vertical collaboration arises when shippers and customers collaborate to help each other optimize their objective, while horizontal collaboration takes place when shippers collaborate among them (and/or the same do customers) at the same logistic level. Ergun et al. [37] develop a collaboration model among shippers, involving only full truckload companies, to identify tours that minimize asset repositioning costs. The same authors discuss in [36] how to reduce truckload transportation cost through the identification of repeatable, dedicated continuous move tours using collaboration among carriers to reduce the need for repositioning and lowering costs. Mason et al. [61] focus on customer driven supply chain and freight management with the aim of studying if collaborative models for management transportation give optimized solutions.

Some authors addressed carrier collaboration from a perspective of costs and profits allocations, possibly within a game theory context. Figliozzi [40] proposes a setting in which a set of carriers, each with its own customers, has some incentive to submit all customers requests to a centralized collaborative decision making mechanism based on sequential second-price auction. Ozener et al. [66] focus, instead, on reducing costs through collaboration. Given a set of lanes carriers have to serve, their aim is to set up a process to exchange lanes either sharing or not sharing information about customers and/or side payments. Argawal and Ergun [1] study transportation networks that operate as an alliance among different carriers. They focus on formation of alliances and network design using both mathematical programming and game theory to investigate the mechanism that leads to an optimal collaborative strategy. In contrast to those studies, in our setting we deal with a network of carriers (regional or functional) that form a coalition to collaborate and we consider as a given fact that collaboration is better than competition, as pointed out in Argawal [2], Meyer [62], and Fugate [43].

Audy et al. [11] and Krajewska et al. [54] are case oriented papers. The former deals with the supply chain of the Canadian furniture industry, while the latter deals with more general coalitions among carriers. Both make use of game theory to allocate cost among companies, customers, carriers and coalitions. In particular, in [54] the authors also use the classic Shapley value to allocate costs among carriers and coalitions of carriers. In [45] various criteria are presented to allocate costs using classical game theory in a vehicle routing problem. Our perspective in this chapter is quite different. While we do not focus on cost allocation among carriers, we study how to improve profits for the whole carriers network within the framework of a fixed collaboration agreement by stating our model as a prize-collecting arc routing problem with several carriers and depots.

Since the CUARP belongs to the class of arc routing problems with profits, we next recall some relevant literature related to this class. Christofides et al. [19] present a directed version of the Rural Postman Problem (RPP) [56], which is later generalized as the Directed Profitable RPP (DPRPP) in [7], where some arcs may not be served by paying a penalty for each of them. Aráoz et al. [6] propose the Privatized Rural Postman Problem (PRPP) whose objective is to find a tour maximizing the profit gained, starting and ending at a fixed depot. Different variants of this problem were proposed by several authors. We mention the Clustered Prize-Collecting Arc Routing Problem introduced by Aráoz et al. [4] and its windy version studied by Corberán et al. [23]. For a comprehensive survey on arc routing with profits we refer to Archetti et al. [8]. Differently from the above studies, we focus on a multi-depot model to optimize collaboration among carriers.

3.3 Collaboration Uncapacitated Arc Routing Problem

The CUARP can be stated as follows. We consider a set of carriers, each with one depot and one vehicle. We assume that the problem is uncapacitated and do not consider capacity constraints on vehicles. Customers are represented as arcs of a graph and are served when the vehicle traverses the corresponding arcs. Carriers reach a collaboration agreement, described in the following, under the guidance and surveillance of a third party central decision maker. The goal is to find one route for each carrier, in the framework of the collaboration agreement, such that the profit is maximized. The collaboration scheme that we study is the following. Each customer is associated with a specific carrier. Each carrier partitions its customers in two sets:

- customers the carrier must serve because of contractual obligations or other types of considerations, such as relevance or convenience;
- customers the carrier is willing to share with other carriers, because of a low level of geographical synergies with other customers or a low profitability.

The customers of the first type are called *required* and form the *required set*, whereas the customers of the second type are called *shared* and form the *shared set*. Required and shared customers are called *demand* customers. We will refer both to customers associated with (or assigned to) carriers and to carriers associated with (or assigned to) customers. We note that each customer is assigned to one carrier, whereas there are usually several customers assigned to one carrier. While required customers must be served by their associated carrier, shared customers can be served by any carrier. We allow a shared customer not be served by any carrier of the coalition. This corresponds to the situation where the customer is not profitable for any carrier and, in the later stage, a carrier that does not belong to the coalition will be searched.

Each customer, if served, will pay an amount of money to its associated carrier. For each shared customer the associated carrier will share part of this revenue with the carrier that will end up actually serving the customer. Carriers determine the side payment for each shared customer for the case it will be served by a different carrier.

Each customer can be served at most once and by only one carrier. Thus, the revenue is collected only the first time the corresponding arc is traversed, even if the arc is traversed more than once. If a shared customer is not served, a penalty is charged to the associated carrier. Every time an arc is traversed a cost is charged, independently of whether or not it corresponds to a demand arc.

We identify carriers with vehicles, depots, and routes, and assume that routes start and end at the same depot.

In order to state the CUARP formally, we first introduce some notation. Let $G = (V, A)$ be a strongly connected directed graph with vertex set $V = \{1, \dots, n\}$ and arc set A. When needed, arcs will be denoted by their end-vertices $a = (u, v)$. A non-negative traversal cost c_a is associated with each arc $a \in A$. The subset of demand arcs (customers) is denoted by $D \subset A$, and the subsets of required and shared customers by R and S, respectively. We have $D = R \cup S$ and $R \cap S = \emptyset$. A non-negative value r_a is associated with each demand arc $a \in D$, which represents the money offered by customer a in exchange of service. Furthermore, a non-negative value $g_a \leq r_a$ is associated with each shared arc $a \in S$, which represents the side payment from the associated carrier to the carrier that provides the service to a. A positive value ϕ_a is also associated with each shared arc $a \in S$, which is the penalty that the associated carrier must pay for not serving customer a.

Let $L = \{1, \dots, k\}$ be the index set for the carriers, each of them with a depot located at a vertex of the graph, denoted by $v^l \in V$, $l \in L$. We also use $V^L = \{v^l | l \in L\} \subset V$ to denote the set of all depots and $I^l = L \backslash \{l\}$, for $l \in L$. For each $l \in L$ we denote by D^l the subset of demand customers associated with l, and by $R^l = D^l \cap R$ and $S^l = D^l \cap S$ its associated required and shared customers, respectively. Customers in D^l, R^l, and S^l will be referred to as *l-demand*, *l-required* and *l-shared* customers, respectively. For $l \in L$, $D^l = R^l \bigcup S^l$ and $R^l \bigcap S^l = \emptyset$. We also have, $D = \bigcup_{l \in L} D^l$, $R = \bigcup_{l \in L} R^l$, and $S = \bigcup_{l \in L} S^l$.

We use the following standard notation. For a nonempty proper subset $F \subseteq V$,

$\gamma(F) = \{a = (u, v) \in A \mid u, v \in F\}$, set of arcs with both vertices in F,
$\delta^+(F) = \{a = (u, v) \in A \mid (u \in F, v \notin F)\}$, set of arcs that start in F and end out of it,
$\delta^-(F) = \{a = (u, v) \in A \mid (u \notin F, v \in F)\}$, set of arcs that start out of F and end in it.

Finally, for each $H \subseteq A$ we define $y(H)$ as $\sum_{a \in H} y_a$.

In the CUARP, we impose that for any carrier $l \in L$, each l-required arc $a \in R^l$ is served by carrier l. Instead, an l-shared arc $a \in S^l$ can be served either by carrier l or by a different carrier $h \in I^l$, or not served at all. A customer $a \in D$ offers a non-negative amount of money r_a in exchange of service. Carrier l collects the revenue r_a for each l-required arc $a \in R^l$ as well as for each served l-shared arc $a \in S^l$, even if it is served by a different carrier $h \in I^l$. If carrier h serves an l-shared arc $a \in S^l$, it collects the non-negative side payment g_a from carrier l. Therefore, the side payment g_a is added to the profit of carrier h and subtracted from the profit of carrier l. If an l-shared arc $a \in S^l$ is not served by any carrier, then the revenue r_a is not collected by carrier l, and carrier l gives no side payment g_a to any other carrier. However, in this case carrier l has to pay the penalty ϕ_a. All routes start and end at the depot of their associated carrier. While carriers with a non-empty required set must certainly perform a route, it is possible for a carrier with empty required set to perform no route. If performed, the route of such a carrier will only serve shared arcs. Carrier l pays a cost c_a each time arc $a \in A$ is traversed in its route. The total profit of carrier $l \in L$ is the difference between its total income and its total costs and side payments, including penalties. The aim of the CUARP is to maximize the total profit of the coalition of carriers.

3.3.1 Formulations

To formulate the CUARP we define the following two sets of decision variables, which identify the arcs that are served and traversed by each carrier $l \in L$.

For each $a \in A$, let

$$y_a^l = \begin{cases} 1 & \text{if } a \text{ is served by vehicle } l, \\ 0 & \text{otherwise.} \end{cases}$$

$$x_a^l = \text{number of times vehicle } l \text{ traverses } a.$$

Associated with a solution (x_a^l, y_a^l) we define the following functions for each carrier $l \in L$:

$$C_1^l = \sum_{a \in A} c_a x_a^l, \quad \text{total traveling cost for carrier } l,$$

$$C_2^l = \sum_{a \in S^l} \phi_a (1 - y_a^l), \quad \text{total penalty payed by carrier } l,$$

$$C^l = C_1^l + C_2^l, \quad \text{total cost for carrier } l,$$

$$P_1^l = \sum_{a \in D^l} r_a y_a^l + \sum_{a \in S^l} \left[(r_a - g_a) \sum_{i \in I^l} y_a^i \right], \text{profit collected by carrier } l \text{ from } l\text{-demand customers},$$

$$P_2^l = \sum_{i \in I^l} \left(\sum_{a \in S^i} g_a y_a^l \right), \quad \text{total side payments collected by carrier } l \text{ from other carriers},$$

$$P^l = P_1^l + P_2^l - C^l, \quad \text{total profit of carrier } l.$$

Furthermore, the set of constraints (C) models the collaboration agreement among carriers:

$$(C) \begin{cases} x^l(\delta^+(u)) = x^l(\delta^-(u)) & l \in L,\ u \in V & (3.1) \\ x^l(\delta^+(v^l)) \geq 1 & l \in L \text{ with } R^l \neq \emptyset & (3.2) \\ x^l(\delta^+(v^l)) \geq y_a^l & l \in L \text{ with } R^l = \emptyset,\ a \in S & (3.3) \\ x^l(\delta^+(F)) \geq y_a^l & l \in L,\ F \subset V \setminus \{v^l\},\ a \in \gamma(F) & (3.4) \\ y_a^l = 1, & a \in R^l,\ l \in L & (3.5) \\ \sum_{l \in L} y_a^l \leq 1 & a \in S & (3.6) \\ y_a^l \leq x_a^l & a \in A,\ l \in L. & (3.7) \end{cases}$$

Flow-in flow-out constraints (3.1) guarantee the symmetry of the vertices, because the number of incoming arcs must be equal to the number of outgoing arcs. Constraints (3.2) and (3.3) guarantee that the carriers routes start from their depots. While (3.2) imposes a route to any carrier $l \in L$ with non-empty required set, constraints (3.3) only impose a route to carriers with empty required sets who serve some shared arc. Constraints (3.4) guarantee that the route of each carrier is connected. Given a subset $F \subset V \setminus \{v^l\}$, if an arc $a \in \gamma(F)$ is served, then some arc $b \in \delta^+(F)$ must be traversed at least once. Hence, each carrier travels a connected route because if it serves some arc from a subset of arcs which does not contain the depot then it has to leave the subset. Note, however, that Constraints (3.4) do not prevent subtours containing no served arc. Since such subtours produce no profit, they will never appear in any optimal solution. Constraints (3.4), together with constraints (3.1)-(3.3), also guarantee that each carrier route ends at its depot. Constraints (3.5) force carrier l to serve all l-required arcs, whereas inequalities (3.6) ensure that l-shared arcs are served by at most one carrier. Finally, inequalities (3.7) impose that all arcs served by a given carrier are traversed by that carrier.

We introduce now the mathematical programming formulation for the CUARP, where we maximize the total profit of the coalition of carriers:

$$z_c = \max \sum_{l \in L} P^l \tag{3.8}$$

$$(C) \tag{3.9}$$

$$x_a^l \in \mathbb{Z}^+, \quad a \in A, l \in L; \qquad y_a^l \in \{0,1\}, \quad a \in D, l \in L. \tag{3.10}$$

Remark 3.3.1: *In the CUARP carriers exchange side payments corresponding to shared arcs served by carriers different from the ones they are assigned to. This means that if, for a given arc $a \in S^l$, some carrier $i \in I^l$ receives g_a for serving l-shared arc a, then carrier l receives $r_a - g_a$. Thus, the profit collected by all carriers is:*

$$\sum_{l \in L} (P_1^l + P_2^l) - C^l = \sum_{l \in L} \left(\sum_{a \in D^l} r_a y_a^l + \sum_{a \in S^l} \left[(r_a - g_a) \sum_{i \in I^l} y_a^i \right] + \sum_{i \in I^l} \sum_{a \in S^i} g_a y_a^l \right) - \sum_{l \in L} C^l =$$

$$\sum_{l \in L} \left(\sum_{a \in D^l} r_a y_a^l + \sum_{a \in S^l} r_a \sum_{i \in I^l} y_a^i - C^l \right)$$

where the last equality follows as the following two sums cancel out since:

$$\sum_{l \in L} \sum_{a \in S^l} g_a \sum_{i \in I^l} y_a^i = \sum_{l \in L} \sum_{i \in I^l} \sum_{a \in S^i} g_a y_a^l.$$

Hence, the objective function (3.8) *can be reformulated as:*

$$\max \sum_{l \in L} \left(\sum_{a \in D^l} r_a y_a^l + \sum_{a \in S^l} r_a \sum_{i \in I^l} y_a^i - C^l \right). \tag{3.11}$$

Therefore, the optimal solution to a CUARP instance is independent of the side payments g_a, $a \in S$, since (3.11) *does not depend on the side payments g_a, $a \in S$, and the domain* (3.1)-(3.7) *is independent of the side payments as well. Thus, we have that an optimal solution to a CUARP instance with side payments g_a, $a \in S$ is also optimal to a CUARP instance with side payments g'_a, $a \in S$, if all other data remain unchanged.*

In the CUARP we can force carriers not to collaborate by adding a constraint that prevents carrier l from serving arcs outside its demand set:

$$y_a^i = 0, \quad a \in S^l, i \in I^l.$$

We define the CUARP *without collaboration*, that we denote as n-CUARP, as the problem obtained by simply adding the above constraints to the CUARP formulation:

$$z_{nc} = \quad \max \sum_{l \in L} P^l \tag{3.12}$$

$$(C) \tag{3.13}$$

$$y_a^i = 0 \qquad a \in S^l, i \in I^l \tag{3.14}$$

$$x_a^l \in \mathbb{Z}^+, \quad a \in A, l \in L; \qquad y_a^l \in \{0,1\}, \quad a \in D, l \in L. \tag{3.15}$$

Observe that the CUARP does not guarantee any profit balance among carriers, possibly limiting the interest for carriers to collaborate. Let us consider, for example, the case of a carrier that has an associated customer that generates little profit because it is not very conveniently located. If the carrier decides to share this customer, the customer might be end up being served by another carrier, for which it is more conveniently located. However, the carrier that decided to share it may simply loose the little profit of the shared customer without gaining anything. In the CUARP the largest carriers will tend to benefit from collaboration more than the small ones.

Below we introduce a variant of the CUARP, that we call the t-CUARP, in which a minimum profit threshold t^l is guaranteed for any carrier $l \in L$. Each carrier may set the threshold to avoid reducing its profit because of collaboration. The threshold for carrier l might be set to be its profit in the n-CUARP. The resulting formulation for the t-CUARP is:

$$z_t = \max \sum_{l \in L} P^l \tag{3.16}$$

$$(C) \tag{3.17}$$

$$P^l \geq t^l \qquad l \in L \tag{3.18}$$

$$x_a^l \in \mathbb{Z}^+, \quad a \in A, l \in L; \qquad y_a^l \in \{0, 1\}, \quad a \in D, l \in L. \tag{3.19}$$

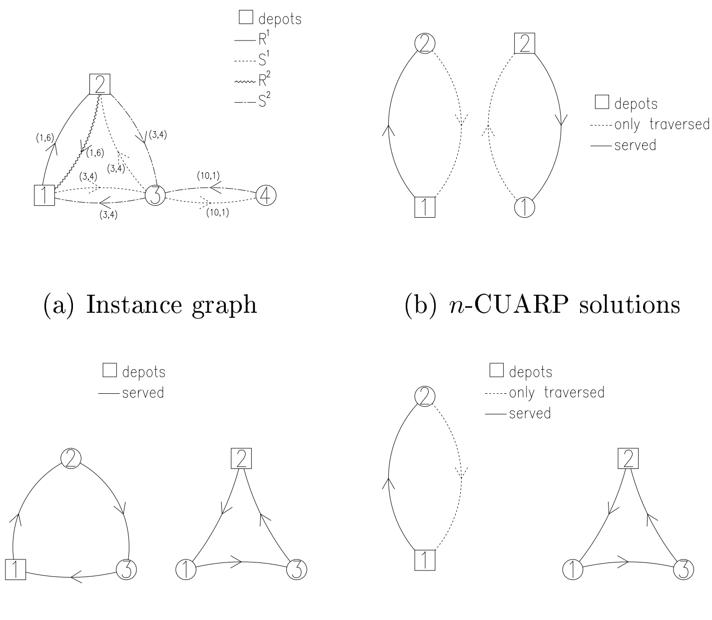
In the t-CUARP we maximize the total profit of the carriers coalition (3.16) as long as the profit of each carrier is not smaller than its threshold (see (3.18)). In contrast to the CUARP, introducing profit thresholds in the t-CUARP gives to side payments g_a a central role, since constraints (3.18) depend on their values. Indeed, it may now happen that a solution which is feasible for the CUARP is no longer feasible for the t-CUARP, because the amount of side payments from a carrier to the others may cause its profit to fall below the given threshold.

We illustrate the behaviour of the different models on an example.

In Figure 3.1(a) a small instance is shown. We consider a graph with 4 vertices $\{1, 2, 3, 4\}$*, whose arcs are partitioned as follows:*

$$R^1 = \{(1, 2)\} \qquad R^2 = \{(2, 1)\}$$

$$S^1 = \{(1, 3); (3, 2); (3, 4)\} \qquad S^2 = \{(2, 3); (3, 1); (4, 3)\}.$$



(a) Instance graph (b) n-CUARP solutions

(c) CUARP solutions (d) t-CUARP solutions

Fig. 3.1: Example

We have only two depots in 1 *and* 2*, respectively. Penalties are set to* 1 *for each shared arc. Each arc in Figure 3.1(a) has a label with two numbers, the first one is the traversing cost and the second one is the profit for serving it. We set the following side payments for serving shared arcs as follows:*

$$g_{(2,3)} = g_{(3,1)} = g_{(4,3)} = g_{(3,4)} = 1, \quad g_{(1,3)} = g_{(3,2)} = 4.$$

In Figure 3.1(b) we represent the solutions of the CUARP without collaboration. Carrier 1 *serves its required customer* $(1,2)$ *in the route 1–2–1, with a profit of* 4*, and pays the penalty for its unserved shared arcs* $(1,3), (3,2)$ *and* $(3,4)$ *with a total penalty of* 3*. Carrier* 2 *serves its required customer* $(2,1)$ *in the route 2–1–2, with a profit of* 4*, and pays the penalty for its unserved shared arcs* $(2,3), (3,1)$ *and* $(4,3)$ *with a total penalty of* 3*. The total profit for the coalition of carriers* 1 *and* 2 *is* 2*. In Figure 3.1(c) the solutions of the CUARP are shown. The route of carrier* 1 *is 1–2–3–1. Its profit is* $P^1 = 0$*. Similarly, the route of carrier* 2 *is 2–1–3–2, with a profit* $P^2 = 12$*. Thus, the collaboration between the* *carriers leads to a total profit of* 12*, with a profit increase of* 83.33% *with respect to the solution of the CUARP without collaboration. Note that the profit is totally gained by carrier* 2*, while carrier* 1 *has a null profit* ($P^1 = 0, P^2 = 12$)*. This allocation of the profit in the coalition of carriers is due to the side payments exchanges between carriers. According to this, carrier* 1 *has no incentive to collaborate with carrier* 2*, as this would result in a profit decrease for carrier* 1*. Observe that the CUARP solution shown in Figure 3.1(c) is not feasible for the t-CUARP instance with profit thresholds set to the individual profits of the n-CUARP (*i.e. $t^1 = t^2 = 1$)*. Figure 3.1(d) shows the optimal routes for this t-CUARP instance, which are 1–2–1 and 2–1–3–2, for carriers 1 and 2, respectively. As in the n-CUARP solution, carrier* 1 *only serves its required arc but no shared arc. On the contrary, carrier* 2 *serves not only its required arc, but also the 1-shared arcs* $(1,3)$ *and* $(3,2)$*, as in the CUARP solution. Less shared arcs are served with respect to the CUARP. Now, the profit of carriers* 1 *and 2 are* $P^1 = 3$ *and* $P^2 = 4$*, respectively. Hence, the total profit of the two carriers is equal to* 7 *which means a profit improvement of* 71.43% *with respect to the n-CUARP and a decrease of* 41.66% *with respect to the CUARP solution. However, the profit coming from collaboration is shared between carriers in a fairer way because of the profit thresholds. In Table 3.1 we summarize profit sharing for the different models in the Example 3.3.1.*

Tab. 3.1: Summary of Example 3.3.1

carrier	n-CUARP	CUARP	t-CUARP
1	1	0	3
2	1	12	4

It may be expected that, in the t-CUARP, a higher value of the side payments results in an increase in the number of arcs served by the coalition. In the following example, we illustrate a counter-intuitive behaviour of the model.

In this example we illustrate the difference between the CUARP and the t-CUARP and the effect of side payments in the t-CUARP. Consider a CUARP instance defined on the same directed graph of Example 3.3.1 with the same two carriers and the same sets of required and shared arcs *for each carrier. Suppose* $\phi_a = 0$ *for all* $a \in D$*, with the following values for the profits and costs:* $r_{12} = r_{21} = 6$*;* $r_{13} = r_{32} = 4$*;* $r_{23} = r_{31} = r_{34} = r_{43} = 1$*;* $c_{12} = c_{21} = 1$*;* $c_{13} = c_{32} = c_{23} = c_{31} = 3$*;* $c_{34} = c_{43} = 10$*. In the optimal* n*-CUARP solution carrier 1 serves its required customer (1,2) in the route 1-2-1, with a profit of 4. Similarly, carrier 2 serves its required customer (2,1) in the route 2-1-2, with a profit of 4. When collaboration is allowed, in the optimal CUARP solution the routes for carriers 1 and 2 become 1-2-1 and 2-1-3-2, respectively. If we set* $g_a = \beta r_a$ *for all* $a \in D$ *with* $\beta = 1$*, we have the following distribution of profits: Carrier 1 gains* $P^1 = 4$ *and Carrier 2* $P^2 = 7$*. If we set* $\beta = 0.5$*, profits become* $P^1 = 8$ *and* $P^2 = 3$*. When we consider the* t*-CUARP with profit thresholds set on the individual profits of the* n*-CUARP, with* $t^4 = t^2 = 4$*, we observe that the solution with* $\beta = 1$ *is still feasible, while that with* $\beta = 0.5$ *is no longer feasible for the* t*-CUARP. For the case* $\beta = 0.5$*, the optimal* t*-CUARP solution keeps unchanged the route of carrier 2 and assigns to carrier 1 the route 1-2-3-1. Hence, the total profit of carrier 1 is* $P^1 = 4$ *and that of carrier 2 is* $P^2 = 4$*. Comparing the* t*-CUARP solutions with* $\beta = 1$ *and* $\beta = 0.5$ *we note that in the former case 2 shared arcs are served while in the latter 4 shared arcs are served. Counter-intuitively, the percentage of shared arcs increases when decreasing* β*. In Table 3.2 we summarize profit sharing for the different models in the Example 3.3.1.*

Tab. 3.2: Summary of Example 3.3.1

	β *independent*	$\beta = 1$		$\beta = 0.5$	
carrier	n-CUARP	CUARP	t-CUARP	CUARP	t-CUARP
1	4	4	4	8	4
2	4	7	7	3	4

3.4 Theoretical results

In this section we present some theoretical results for the CUARP.

3.4.1 Reduction to other problems and complexity

We analyze two particular cases of the CUARP with one single carrier:

- If the shared set S is empty, then $D = R$ and, thus, the single carrier CUARP reduces to the Directed RPP (DRPP). Since the DRPP is known to be NP-Hard (see Lenstra and Rinnooy Kan [56]), also the CUARP is.
- If the required set R is empty, then $D = S$ and, thus, the single carrier CUARP reduces to the DPRPP. Since the DPRPP is NP-hard (see Archetti et al. [7]), this is an alternative proof that CUARP is NP-Hard.

Thus, we can reduce the single carrier CUARP to other problems by changing the size of the shared and required sets. At one extreme, with no shared arcs, we have the DRPP, whereas on the other one, with no required arc, we have the DPRPP.

3.4.2 Impact of collaboration on profit

Remark 3.4.1: Let $\mathcal{I}$ be a CUARP instance and z_c, z_{nc} be the CUARP optimal value and the n-CUARP optimal value over $\mathcal{I}$, respectively. Then, $z_{nc} \leq z_c$.

This result holds trivially, since any feasible solution to the n-CUARP is a CUARP feasible solution.

Remark 3.4.2: Let $\mathcal{I}^1$, $\mathcal{I}^2$ be two CUARP instances and z_{c_1}, z_{c_2} their CUARP optimal values. Suppose these instances differ only because the shared set of the first one is contained in the shared set of the second one and vice versa for the required sets. Then, $z_{c_1} \leq z_{c_2}$.

This result holds trivially, since any feasible solution for $\mathcal{I}^1$ is a feasible solution for $\mathcal{I}^2$. We may say that the attractiveness of collaboration increases by increasing the shared arcs or by decreasing the required arcs.

Remark 3.4.3: Let $\mathcal{I}$ be a CUARP instance, and let z_t, z_{nc} be, respectively, the optimal t-CUARP and n-CUARP values over $\mathcal{I}$, when the t-CUARP thresholds t^l, $l \in L$, are set to the carriers profits of the n-CUARP. Then, $z_{nc} \leq z_t$.

The above result follows, since for instance $\mathcal{I}$ the optimal solution to the n-CUARP is also feasible for the t-CUARP with the given threshold values.

Proposition 3.4.1: Let z_c and z_{nc} denote the optimal values of the CUARP and the n-CUARP over a given instance $\mathcal{I}$, respectively. There exists no finite upper bound for the profit increase ratio $\frac{z_c}{z_{nc}}$.

Proof: Consider a CUARP instance defined on the same directed graph of Example 3.3.1 with the same two carriers and the same sets of required and shared arcs for each carrier. Let $K > 0$ and $\varepsilon > 0$ be two given values and suppose $c_a = K$, for all $a \in A$, and $\phi_a = \varepsilon$, $g_a = K$, for all $a \in D$. Let us also suppose the profits are the following: $r_{12} = r_{21} = 2K + 4\varepsilon$; $r_{13} = r_{31} = r_{23} = r_{32} = 2K - \varepsilon$; $r_{34} = r_{43} = K$.

For this instance, in the optimal n-CUARP solution carrier 1 serves its required customer $(1,2)$ in the route 1–2–1, with a profit of 4ε, and pays the penalty for all its unserved shared arcs (1,3), (3,2) and (3,4) with a total penalty of 3ε. Similarly, in the optimal n-CUARP solution, carrier 2 serves its required customer $(2,1)$ in the route 2–1–2, with a profit of 4ε, and pays the penalty for all its unserved shared arcs (2,3), (3,1) and (4,3) with a total penalty of 3ε. Hence, for this instance $z_{nc} = 2\varepsilon$.

When collaboration is applied, in the optimal CUARP solution carrier 1 serves its required customer (1, 2) and the 2-shared customers (2, 3) and (3, 1) in route 1–2–3–1, whereas carrier 2 serves its required customer (2,1) and the 1-shared customers (1, 3) and (3, 2) in route 2-1-3-2. Demand customers (3,4) and (4,3) remain unserved. The profit of both carrier 1 and carrier 2 is, thus, $(K+4\varepsilon)+(K-K)+(K-K)+(2K-\varepsilon-K)+(2K-\varepsilon-K)-\varepsilon = 3K+\varepsilon$. Hence, for this instance $z_c = 6K + 2\varepsilon$.

Therefore, the profit increase ratio is $\frac{z_c}{z_{nc}} = \frac{6K+2\varepsilon}{2\varepsilon}$, which tends to ∞ either when $K \to \infty$ or when $\varepsilon \to 0$. □

Observe that the optimal solution of the CUARP instance built in the above proof is also feasible for the t-CUARP when the threshold of each carrier is set to its profit without collaboration, *i.e.*, $t^1 = t^2 = \varepsilon$. Therefore the following result also holds:

Corollary 3.4.2: Let z_t and z_{nc} denote the optimal values of the t-CUARP and the n-CUARP over a given instance $\mathcal{I}$, respectively. There exists no finite upper bound for the profit increase ratio $\frac{z_t}{z_{nc}}$.

3.4.3 Game theory results

We introduce here some basic concepts and definitions of cooperative game theory (see Driessen [34] for a comprehensive survey), and relate them to the CUARP. A *cooperative game* consists of a finite number n of players and a characteristic function. Players can form different coalitions in order to achieve a better result in the game. The coalition of all players is called *grand coalition*. The *characteristic function* $v : 2^n \to \mathbb{R}$ is a function from the set of all possible coalitions of players to a set of payments such that $v(\emptyset) = 0$. Each coalition is associated with a payment and each player has its own share of payment. In general, there are many different payments allocations suitable as solutions for a cooperative game. Hence, a key concept in cooperative game theory is the *core*, which is formed by those payments allocations $y_1, \dots y_n$ that satisfy the following conditions:

$$\sum_{j \in S} y_j \geq V(S), \text{ with } S \subset N,$$
$$\sum_{j \in N} y_j = V(N).$$

The former conditions prevent players from colluding to form a subcoalition in order to gain more. In the case $S = \{j\}$, the condition ensures that each player receives at least what he could get on his own, and is called *individual rationality condition*. The latter condition implies that the allocations $y_1, \dots y_n$ split the total value gained by the grand coalition. This condition is called *budget balance* or *efficiency condition*. Given the above two conditions, no player has an advantage by leaving the grand coalition and the profit allocation is called *stable*.

We consider now a cooperative game based on the CUARP, using its mixed integer programming formulation as characteristic function, as suggested by Gothe et al. in [45]. Carriers play the role of game players. The CUARP and the t-CUARP allocate profit to the players. Both profit allocations fulfill the budget balance/efficiency condition because the sum of the profits of the carriers equals the maximum attainable profit of the whole coalition. However, the CUARP breaks the individual rationality condition because it may happen that a carrier gains more on its own without collaborating, as in Example 3.3.1. On the other hand, the t-CUARP with the n-CUARP profits as thresholds fulfills the individual rationality condition. Hence, the solution of the 2 carrier t-CUARP belongs to the core of the game when we set the n-CUARP profits as thresholds. In this case, the core is always non empty if the instance is feasible. In contrast, we cannot assure that the solutions of the t-CUARP with more than 2 carriers or with general thresholds belong to the core of the game. Even if the thresholds are set to the value of individual profit without collaboration, 2 or more carriers may collude to achieve a greater profit than that of the grand coalition. We might ensure that the t-CUARP profit allocation belongs to the core of the game, by adding a new set of constraints to the t-CUARP, imposing for each subset of carriers a minimum profit of at least its CUARP profit. However, adding this new set of constraints may cause the instance to become infeasible and the core of the cooperative game be empty. In a set of experiments this is what happened. Either the core was empty or the solution was identical to the solution without collaboration. Hence, we solved the t-CUARP adding only individual thresholds which are those that directly matter to the carriers.

3.5 The branch-and-cut algorithm

Inequalities (3.4) impose the connectivity of the route associated with each carrier with its depot. As the number of such constraints is exponential in the number of vertices in the input graph, $|V|$, we use a separation procedure that allows us to incorporate them in the formulation only when needed. Next, we describe the separation algorithm that we used for the exact solution of the CUARP and of the t-CUARP. The separation procedure uses as input vectors $\hat{x} = (\hat{x}^l)_{l\in L} \in R^{|A|\times|L|}$ and $\hat{y} = (\hat{y}^l)_{l\in L} \in R^{|A|\times|L|}$ satisfying constraints (3.1)–(3.3) and (3.5)-(3.7). The output of the algorithm will indicate whether or not there exists some inequality (3.4) violated by $\hat{x}$ and $\hat{y}$. In this case the algorithm will return one such inequality, *i.e.* the index of a carrier $l \in L$, a set $F \subset V \setminus \{v^l\}$ and an arc $a \in \gamma(F)$ such that the corresponding constraint (3.4) is violated by $\hat{x}$ and $\hat{y}$.

For each carrier $l \in L$, let $G^l(\hat{x}^l) = (V^{\hat{x}^l}, A^{\hat{x}^l})$ denote the support graph of $\hat{x}^l$, with $A^{\hat{x}^l} = \{a \in A \mid \hat{x}^l_a > 0\}$, and $V^{\hat{x}^l} = V(A^{\hat{x}^l})$ the subset of vertices incident with some arc of $A^{\hat{x}^l}$. To separate inequalities (3.4), associated with a given carrier $l \in L$, we consider the support graph $G^l(\hat{x}^l)$. If $G(\hat{x}^l)$ is not connected, then each connected component with vertex set $C \subseteq V^{\hat{x}^l} \setminus \{v^l\}$ such that $\hat{y}(\gamma(C))>0$ defines a violated constraint (3.4) for carrier l, since $\hat{x}^l(\delta^+(C)) = 0$ and $\hat{y}^l_a > 0$, for some $a \in \gamma(C)$. If $G(\hat{x}^l)$ is connected, we compute the tree of min-cuts $T^l(\hat{x})$ relative to the capacities vector $\hat{x}^l$ (see, for instance, [44, 46]). Then, we use an adaptation of the algorithm of Belenguer and Benavent [15]. For each min-cut $\delta^+(F)$, $v^l \notin F$, represented in $T^l(\hat{x})$, we identify the arc $\hat{a} \in \gamma(F)$ of maximum value, *i.e.* $\hat{a} \in \arg\max\{\hat{y}^l_a \mid a \in \gamma(F)\}$. Now, if $\hat{x}^l(\delta^+(F)) < \hat{y}^l_{\hat{a}}$, the connectivity constraint (3.4) associated with l, F and $\hat{a}$ is violated by $\hat{x}^l$ and $\hat{y}^l$. The above separation is exact and similar to the procedure used by other authors to separate connectivity constraints similar to (3.4) for other arc routing problems [16, 3, 4, 23].

The computational complexity of the above algorithm is dominated by that of the algorithm to obtain the min-cut tree associated with each carrier, which is $\mathcal{O}(|V|^4)$ as pointed out in [44] and in [46].

3.6 Data generation and computational results

We present in this section the numerical results obtained on a series of computational experiments. Programs were coded in Java using CPLEX 12.5 library (64 bit) for the solution of the mixed integer problems. Default parameters were used. All tests were run on a HP Z400 Workstation, 64 bit, 3.33 GHz, 12.0 RAM. Since there are no available CUARP benchmark instances, we generated instances from the 118 PRPP benchmark instances used in [5]. These PRPP instances were derived from well-known RPP instances, which are divided in five groups. The first group contains two data sets, A and B, obtained from the Albaida Spain Graph (see Corberán and Sanchis [24]). The second group contains the 24 instances (labeled P) of Christofides et al. [19]. The last three groups contain instances from Hertz et al. [49]: 36 instances with vertices of degree 4 and RPP disconnected required edge sets (labeled D), 36 grid instances (labeled G), and 20 randomly generated instances (labeled R). Below we explain how the remaining data of the instances were defined. First, the original undirected graph is transformed in a directed graph in the following way.

- All arcs are defined from edges of the original graph (see [5]).
  - Each original edge is transformed in two arcs with probability 0.1, and in one single arc with probability 0.9. In the latter case, the direction of the arc is randomly chosen with equal probability.
  - Arcs inherit their costs from the original edges. When two arcs are generated from the same edge, both arcs have the same cost.
- If needed, when all original edges have been considered, additional arcs are defined to guarantee that the resulting graph is strongly connected.
  - For each pair of vertices, $u, v \in V$, for which the directed graph defined earlier contains no path from u to v, we define a non-demand arc (u, v) and assign to it the cost of the shortest path in the undirected graph connecting u and v.
- Demand arcs are selected starting from the required edges of [5], as follows:
  - If the original edge is a demand edge that has been transformed in two arcs, both transformed arcs become demand arcs with prob-

ability 0.15. Otherwise, one transformed arc is randomly selected as demand arc while the other one becomes non-demand.

- If the original edge is a demand edge that has been transformed in one single arc, the transformed arc becomes a demand arc.
- If the original edge is non-demand the transformed arc(s) is(are) non-demand.

- The profit of each demand arc $a \in D$ is defined as $r_a = 2b_e$, where b_e is the profit of the undirected edge in the corresponding PRPP instance [5]. If e is a required edge of the RPP instance, b_e is a number randomly generated from an integer uniform distribution in the range $[c_e,\ 3c_e]$. Otherwise, $b_e = 0$.

- The side payment of each demand arc $a \in D$, g_a, is set to a fraction β of its profit, *i.e.*, $g_a = \beta r_a$. For each instance, we use two values for β, namely 0.5 and 1.

To define the demand sets, we denote by P_{uv} the shortest path from u to v in the directed graph and by $c(P_{uv})$ its cost. Then, we define the following auxiliary set for each carrier $l \in L$:

$$\bar{D}^l = \{a \in D | w_a \leq 0.75 \wedge c(P_{v^l u}) \leq c(P_{v^i u}), \quad \text{for all } i \in I^l\},$$

where u denotes the end-vertex of arc a closest to v^l and w_a a random number in $[0, 1]$ associated with arc a. Since the sets $\bar{D}^l$ are not necessarily disjoint for different carriers the demand sets are finally defined as

$$D^1 = \bar{D}^1, \qquad D^l = \bar{D}^l \setminus \left(\bigcup_{k=1}^{l-1} \bar{D}^k \right), l \in \{2, \ldots, |L|\},$$

Broadly speaking, we assign demand arcs to the carrier with smallest distance from/to its depot with a probability of 75%. Moreover, if through this procedure a required arc belongs to more than one $\bar{D}^l$ set, then it is assigned to the required set of the smallest index carrier, among the ones whose $\bar{D}^l$ sets contain the arc.
To divide each carrier demand set into shared and required sets, firstly, we list all demand arcs for each carrier according to their distance from the depot. Then, the less distant arcs form the required set while the remaining

ones the shared set. Shared and required sets are defined to ensure that at least a percentage p of the demand arcs are shared. We fix p equal to 50%. We denote with M the vertex most distant from vertex 1, and with m the vertex whose distance from 1 is the median of all distances from 1 to $v \in V$.

We set the same penalty value for all the shared arcs associated with the same carrier. For each carrier $l \in L$, this penalty is set to the average profit loss per unserved l-shared arc. That is, for $l \in L$, $a \in S^l$,

$$\phi_a = \begin{cases} \max\left\{0, \left\lfloor \sum_{a \in S^l} \frac{r_a - c_a}{|S^l|} \right\rfloor\right\} & \text{if } S^l \text{ is not empty,} \\ 0 & \text{otherwise.} \end{cases}$$

The minimum threshold values t^l, $l \in L$, are given by the profits of the n-CUARP. The instances are available at the following link: `http://or-brescia.unibs.it/instances`.

Table 3.3 summarizes information on the instances, which have been grouped according to their characteristics and to their sizes. These instances were generated with 2 depots located at $v^1 = 1$ and $v^2 = M$. Columns under *#inst.* and *#vertices* give, respectively, the number of instances in the group and the number of vertices of the instances in the group. Columns under $|R^1|$ and $|R^2|$ give the number of 1-required and 2-required arcs, respectively. Similarly, columns under $|S^1|$ and $|S^2|$ give the number of 1-shared and 2-shared arcs.

Tab. 3.3: Instance summary

	#inst.	*#vertices*	$\lvert R^1\rvert$	$\lvert R^2\rvert$	$\lvert S^1\rvert$	$\lvert S^2\rvert$
AA	1	102	48	34	48	35
AB	1	90	50	23	50	23
P	24	7-50	0-48	2-46	1-48	3-47
D16	9	16	5-8	7-10	5-8	8-11
D36	9	36	15-21	16-21	15-21	17-21
D64	9	64	32-41	20-32	33-42	21-33
D100	9	100	48-67	32-50	49-67	33-50
G16	9	16	0-3	0-3	1-4	1-3
G36	9	36	1-9	3-11	1-9	3-12
G64	9	64	2-13	6-21	3-14	6-22
G100	9	100	8-29	11-32	8-29	12-33
R20	5	20	10-23	5-15	10-24	6-16
R30	5	30	19-37	14-25	19-37	14-26
R40	5	40	25-56	16-47	26-57	17-48
R50	5	50	33-57	26-56	33-58	26-56

Table 3.4 summarizes the information on the values of the parameters that have been used and the number of tested instances in each case. The complete set of instances consists of 971 instances.

Tab. 3.4: Number of instances for each combination of parameters

Location of depots	CUARP	n-CUARP	CUARP with different proportions of shared/required arcs	t-CUARP	
				β	
				0.5	1
$v^1 = 1$, $v^2 = M$	118	118	27 (G16)	118	118
$v^1 = 1$, $v^2 = m$	118	118			
$v^1 = 1$, $v^2 = m$, $v^3 = M$	118	118			

A first set of experiments was run on the set of instances with two carriers, *i.e.* $L = \{1, 2\}$, with depots v^1 and v^2 located at vertices 1 and M, respectively. The results for the CUARP are summarized in Table 3.5. Columns under CUARP and n-CUARP give the average net profit of carriers for the CUARP and the n-CUARP, respectively. Columns under $\#S1$ and $\#S2$ give the average number of shared arcs served by each carrier, while the column under $\%serv.$ gives the average percentage of shared arcs served in total $(100\frac{\#S1+\#S2}{|S^1|+|S^1|})$. Columns under $\#S12$ and $\#S21$ give, respectively, the average number of 1-shared arcs which are served by carrier 2, and vice versa. The column under $\%exc.$ gives the average percentage of arcs exchanged between carriers (*i.e.* the average percentage of shared arcs that are served by a carrier different from the one they are assigned to, $100\frac{\#S12+\#S21}{\#S1+\#S2}$). Column under *increase* gives the average percentage profit increase due to collaboration $(100|\frac{z_c - z_{nc}}{z_c}|)$. Finally, the last column under *time* gives the average computing times in seconds.

The results of Table 3.5 illustrate the positive effect of the carriers collaboration. The average percentages $\%serv.$ and $\%exc.$ range in [67.0, 100.0] and in [41.0, 76.7], respectively. This means that there is a reasonable percentage of shared arcs served and exchanged between carriers. The average percentage profit increase of CUARP with respect to n-CUARP ranges in [0.7, 41.1]. In general, the average CPU time required to solve small instances, such as those of P, D16, D36, G16, R20, R30, R40 groups, is less than 1 second. Solving the remaining instances required a little bit more computational effort, which pushed the CPU time up to a max of 158.0 seconds. The average CPU time is equal to 8.3 seconds. In Tables 3.6 and 3.7 we summarize the results for the t-CUARP, with $\beta = 0.5$ and $\beta = 1$, respectively. We tested separately the instances with these two different values of β, which affect the side payments and, thus, the profit threshold constraints. In these tables we have the additional columns *increase*1 and *increase*2 that show the profit increase for carrier 1 and 2. Comparing the results in Tables 3.6 and 3.7 with those in Table 3.5 we note that the average profit increase due to collaboration is smaller in t-CUARP than in CUARP. Instead, the average percentage %*serv.* may even increase as, for example, for the instances in the D100 group in both cases with $\beta = 0.5$ or $\beta = 1$. The CPU times for the t-CUARP are comparable to those of the CUARP. We note that in some groups of instances the t-CUARP value is smaller when $\beta = 1$ than when $\beta = 0.5$. For instance, in P %*serv.* increases from 90.5 (for $\beta = 0.5$) to 91.1 (for $\beta = 1$). Still the value of CUARP is greater for $\beta = 0.5$ (392.8) than for $\beta = 1$ (379.2). This t-CUARP behavior was explained in Remark 3.3.1 and illustrated in Example 3.3.1. Table 3.8 shows the characteristics of the instances that are obtained when displacing the depot of the second carrier (v^2) from vertex M to vertex m. Due to how sets S^l and R^l are defined when generating the instances, by displacing v^2 the average number of shared arcs increases and the average number of required arcs decreases. Table 3.9 summarizes the results with this new set of instances. Obviously, results in Table 3.9 cannot be compared with those in the previous tables, since we deal with completely different instances. The CPU times required to solve these new instances are comparable to those required for solving the instances in Table 3.3. Below we describe the results obtained in the experiments with CUARP and 3 depots, located in vertices 1, m and M, respectively. We have been able to solve all 118 instances. In Tables 3.10 and 3.11 we introduced additional columns to deal with 3 depots. In particular, columns under $|R^3|$, $|S^3|$ of Table 3.10 give the number of 3-required and 3-shared arcs, respectively. In Table 3.11 columns under $\#S$ and $\#Sexc$ give, respectively, the average number of shared arcs served and the average number of shared arcs exchanged. In Table 3.11 we note that the range of the average percentage of served shared arcs (%*serv.*) and the average percentage of exchanged shared arcs and (%*exc.*) is [35.0, 58.5] and [26.4, 46.5], respectively. Moreover, the average percentage profit increase ranges in [0.0, 16.8]. However, comparing these results with those for the CUARP with 2 depots given in Table 3.5, we note that increasing the number of carriers from 2 to 3 does not necessarily increase the profit. As an example, for the G16 instances, when adding a new depot, *increase* decreases from 21.5% to 16.8%. The computing times to solve instances with 3 carriers are similar to those required for the instances with 2 carriers. Finally, in order to test the behaviour of the model with different proportions of shared/required sets, some additional computational experiments were run. For these experiments we used the instances in G16 with depot located in 1 and in M. Firstly, we restricted the shared sets ensuring that at least 25% of the demand arcs belong to them. Then we enlarge them, ensuring that at least 75% of the demand arcs to be shared. Finally, we set them all as shared arcs. We performed 27 additional computational experiments. In Table 3.12 we show the profits for each case and each instance. Under columns n-CUARP, CUARP-25, CUARP, CUARP-75, CUARP-100 we give the optimal values for n-CUARP, CUARP with 25% of arcs shared, CUARP, CUARP with 75% and 100% of arcs shared, respectively. We note that profit does not decrease when we increase the number of arcs in the shared sets. This is consistent with Remark 3.4.2. We can conclude that the attractiveness of collaboration increases when we increase the number of shared arcs.

Tab. 3.5: Results for CUARP

	n-CUARP	CUARP	$\#S1$	$\#S2$	$\#S12$	$\#S21$	$\%serv.$	$\%exc.$	$increase$	$time$
AA	14442.0	24509.0	41.0	41.0	28.0	34.0	98.8	75.6	41.1	46.2
AB	1411.0	17911.0	26.0	47.0	16.0	40.0	100.0	76.7	92.1	21.2
P	379.2	392.9	13.4	12.9	8.3	7.6	90.5	54.7	4.0	0.4
D16	1211.8	1220.9	5.9	7.0	3.8	3.4	79.5	56.1	0.8	0.1
D36	2443.9	2460.6	17.1	15.9	9.1	8.3	90.8	52.7	0.7	0.4
D64	3582.6	3616.6	33.3	27.4	15.6	17.2	94.3	53.8	1.1	2.8
D100	4810.6	4873.2	42.7	54.0	22.4	35.8	97.5	60.2	1.5	14.6
G16	10.3	13.0	1.2	2.7	0.6	1.1	94.4	42.5	21.5	0.1
G36	42.8	51.0	4.8	8.0	2.3	3.0	100.0	41.0	16.0	1.4
G64	87.6	103.7	13.0	11.3	8.2	5.3	100.0	54.1	15.9	9.7
G100	139.9	170.8	21.9	18.0	12.4	9.9	100.0	54.9	18.7	70.3
R20	52221.8	53101.8	12.2	7.4	4.2	4.8	72.5	46.0	2.3	0.1
R30	67804.2	69230.2	15.6	15.2	6.0	9.8	62.8	52.2	2.0	0.3
R40	86734.0	90811.8	21.8	31.8	9.2	20.2	73.1	56.6	19.8	0.6
R50	91811.4	96350.4	28.6	33.8	15.4	21.4	67.0	58.1	9.8	1.1
						max	100.0	100.0	92.1	158.0
						$average$	89.5	53.1	9.2	8.3

Tab. 3.6: Results for t-CUARP, $\beta = 0.5$

	n-CUARP	CUARP	#$S1$	#$S2$	#$S12$	#$S21$	%$serv.$	%$exc.$	$increase1$	$increase2$	$increase$	$time$
AA	14442.0	18722.0	42.0	39.0	23.0	28.0	97.6	63.0	22.9	30.3	10.5	58.9
AB	1411.0	11946.0	45.0	26.0	19.0	24.0	97.3	60.6	88.2	107.0	39.4	47.9
P	379.2	392.8	12.3	14.0	7.6	8.0	90.5	53.2	4.0	4.1	3.9	0.5
D16	1211.8	1220.4	5.0	7.9	3.2	3.8	79.5	53.1	0.9	0.7	0.7	0.2
D36	2443.9	2460.6	15.2	17.7	9.4	10.4	90.5	59.6	0.9	0.7	0.7	0.6
D64	3582.6	3616.6	31.0	29.9	16.6	20.6	94.5	61.0	1.0	1.2	1.1	9.3
D100	4810.6	4873.2	48.3	48.7	29.1	37.0	97.9	68.2	2.8	0.8	1.5	21.3
G16	10.3	12.2	1.4	2.4	0.6	0.9	94.4	32.0	18.9	5.7	14.3	0.1
G36	42.8	48.9	3.2	9.4	1.0	3.1	98.4	28.6	9.9	17.4	10.8	1.6
G64	87.6	99.7	12.2	12.1	6.4	4.3	100.0	46.6	12.8	20.6	12.8	11.2
G100	139.9	169.0	21.7	18.2	12.6	10.2	100.0	56.2	16.8	16.7	17.2	76.5
R20	52221.8	53061.8	11.6	8.0	4.6	5.8	72.5	50.7	2.1	3.4	2.1	0.2
R30	67804.2	69230.2	18.0	12.8	6.8	8.2	62.8	46.1	1.2	2.4	2.0	0.4
R40	86734.0	90539.0	24.0	29.6	14.6	23.4	73.1	71.8	4.3	7.7	19.5	0.9
R50	91811.4	96350.4	31.0	31.4	18.2	21.8	67.0	62.8	7.0	10.9	9.8	1.9
						max	100.0	100.0	88.2	107.0	85.7	158.0
						$average$	89.3	52.6	7.3	7.9	7.1	10.4

Tab. 3.7: Results for t-CUARP, $\beta = 1$

	n-CUARP	CUARP	#$S1$	#$S2$	#$S12$	#$S21$	%$serv.$	%$exc.$	$increase1$	$increase2$	$increase$	$time$
AA	14442.0	24509.0	41.0	41.0	28.0	34.0	98.8	75.6	41.1	37.8	41.1	18.0
AB	1411.0	17911.0	26.0	47.0	16.0	40.0	100.0	76.7	92.1	106.7	77.3	13.3
P	379.2	392.5	12.2	14.1	6.9	7.5	91.1	44.0	5.3	2.3	3.5	0.7
D16	1211.8	1219.6	5.7	7.2	3.1	3.0	79.5	46.1	0.6	1.0	0.6	0.3
D36	2443.9	2460.3	14.0	19.0	8.3	10.6	90.8	56.7	0.6	0.9	0.7	0.9
D64	3582.6	3616.6	30.7	30.1	15.9	20.1	94.3	59.3	1.1	0.9	1.1	11.0
D100	4810.6	4873.2	52.4	44.4	29.0	32.8	97.7	63.8	1.6	1.8	1.5	23.3
G16	10.3	11.8	1.8	2.1	0.7	0.7	94.4	30.2	24.2	7.9	12.0	0.2
G36	42.8	44.9	5.4	7.2	1.7	1.6	98.4	21.4	11.1	1.2	4.3	2.3
G64	87.6	96.3	10.6	13.8	4.7	4.2	100.0	36.5	9.1	9.6	9.1	15.4
G100	139.9	155.0	19.7	20.2	9.9	9.6	100.0	48.0	10.9	9.4	11.0	100.2
R20	52221.8	53061.8	12.2	7.4	5.0	5.6	72.5	52.9	1.8	4.4	2.2	0.2
R30	67804.2	69190.2	15.4	15.4	6.8	10.8	62.8	56.1	2.5	1.8	1.9	0.8
R40	86734.0	90539.0	29.0	24.6	15.2	19.0	73.1	62.8	21.0	2.7	19.5	1.1
R50	91811.4	96350.4	32.2	30.2	19.0	21.4	67.0	64.3	11.2	8.9	9.8	2.1
						max	100.0	80.0	100.0	106.7	85.7	291.9
						average	89.5	47.9	8.3	4.9	6.2	12.3

Tab. 3.8: Instance summary after moving depots

	#inst.	#vertices	$\|R^1\|$	$\|R^2\|$	$\|S^1\|$	$\|S^2\|$
AA	1	102	31	49	31	49
AB	1	90	28	45	28	45
P	24	7-50	1-57	2-49	2-57	2-49
D16	9	16	5-9	6-11	6-9	7-11
D36	9	36	16-22	12-22	16-22	12-22
D64	9	64	26-42	22-39	26-42	22-40
D100	9	100	42-50	50-55	42-40	51-56
G16	9	16	0-2	0-4	1-3	0-4
G36	9	36	2-7	3-12	2-8	3-12
G64	9	64	4-20	6-17	5-21	6-17
G100	9	100	8-27	9-31	9-27	10-31
R20	5	20	8-20	8-20	8-20	9-20
R30	5	30	18-30	14-30	19-30	15-31
R40	5	40	18-67	22-48	19-68	23-48
R50	5	50	31-48	35-66	31-49	35-66

Tab. 3.9: Results for CUARP after moving depots

	n-CUARP	CUARP	#S1	#S2	#S12	#S21	%serv.	%exc.	increase	time
AA	23570.0	23596.0	36.0	43.0	24.0	18.0	98.8	53.2	0.1	13.9
AB	17657.0	17693.0	39.0	34.0	26.0	15.0	100.0	56.2	0.2	10.0
P	380.3	390.9	11.2	14.5	6.6	7.3	91.8	54.7	4.3	0.3
D16	1221.2	1225.6	5.9	6.0	4.3	2.6	73.7	57.2	0.0	0.1
D36	1929.8	2072.8	19.7	12.2	10.6	7.0	87.9	54.5	0.1	0.6
D64	3622.3	3654.8	28.7	31.7	15.2	18.4	94.0	55.9	0.0	2.8
D100	5058.0	5106.2	39.6	55.4	24.8	30.2	94.9	58.0	0.0	14.5
G16	9.8	13.1	1.0	2.9	0.4	1.3	92.2	39.0	0.2	0.1
G36	38.7	45.9	4.7	7.8	3.0	3.9	100.0	54.9	0.2	1.8
G64	86.6	102.2	13.4	10.8	6.7	5.4	100.0	50.5	0.2	9.6
G100	147.0	167.6	20.1	20.2	10.8	10.4	100.0	52.3	0.2	103.8
R20	52781.8	53861.2	10.6	7.4	5.2	4.4	66.6	53.7	0.0	0.1
R30	63098.6	64668.2	13.8	20.2	8.4	11.4	68.5	56.2	0.0	0.2
R40	81383.0	83126.0	21.0	29.8	11.0	17.2	67.4	54.6	0.0	0.8
R50	82934.0	88777.4	30.8	36.8	16.6	15.6	71.4	48.2	0.2	1.5
						max	100.0	88.9	0.9	185.1
						average	88.6	53.3	0.1	10.5

Tab. 3.10: Instance summary with 3 depots

	#inst.	*#vertices*	$\|R^1\|$	$\|R^2\|$	$\|R^3\|$	$\|S^1\|$	$\|S^2\|$	$\|S^3\|$
AA	1	102	40	22	17	41	23	17
AB	1	90	46	11	17	46	12	17
P	24	7-50	1-43	1-42	0-22	2-44	1-42	1-23
D16	9	16	2-5	6-10	1-5	2-6	6-11	2-6
D36	9	36	10-18	10-16	7-11	11-18	10-16	7-12
D64	9	64	27-40	8-22	12-18	27-41	9-23	13-18
D100	9	100	36-61	13-40	19-29	36-62	13-40	20-29
G16	9	16	0-3	0-2	0-2	1-3	1-2	0-2
G36	9	36	1-8	2-10	1-6	1-8	2-10	1-6
G64	9	64	2-13	4-14	2-11	2-13	5-15	2-11
G100	9	100	6-23	7-24	5-14	6-24	8-24	5-14
R20	5	20	6-23	3-24	3-14	6-24	4-24	4-14
R30	5	30	6-40	2-24	3-15	6-40	3-24	415
R40	5	40	6-48	2-31	3-27	6-48	3-31	4-415
R50	5	50	6-53	2-44	3-28	6-54	3-45	4-415

Tab. 3.11: Results for CUARP with 3 depots

	n-CUARP	CUARP	*#S*	*#Sexc*	*%serv.*	*%exc.*	*increase*	*time*
AA	25481.0	25513.0	42.0	16.0	51.9	38.1	0.1	41.8
AB	20898.0	20930.0	44.0	19.0	58.7	43.2	0.2	36.7
P	433.8	445.6	14.3	5.4	46.3	39.2	3.9	1.0
D16	1379.2	1388.1	7.7	3.1	46.5	41.2	0.7	0.1
D36	2661.4	2681.4	17.3	7.2	47.4	41.8	1.0	1.0
D64	4068.2	4015.3	36.8	13.6	57.2	36.9	0.6	9.3
D100	5387.2	5438.0	54.6	18.3	54.7	33.7	1.1	55.5
G16	15.6	18.0	2.9	1.3	66.7	42.8	16.8	0.1
G36	52.4	61.3	6.6	1.9	54.5	26.4	15.5	3.7
G64	111.8	121.7	12.2	4.4	49.4	35.2	6.9	24.6
G100	171.6	191.7	21.4	6.7	50.1	28.7	11.3	248.1
R20	60528.6	60537.8	15.4	5.6	58.5	33.0	0.0	0.2
R30	88964.0	88986.0	16.4	5.2	35.0	30.8	0.0	0.5
R40	115561.6	115582.8	34.6	13.8	45.9	35.9	0.0	1.4
R50	128557.6	128587.0	40.4	18.4	43.0	46.5	0.0	3.2
				max	100.0	100.0	80.0	546.3
				average	50.6	36.7	4.9	27.2

Tab. 3.12: CUARP G16 results with increasing size of shared sets

	n-CUARP	CUARP-25	CUARP	CUARP-75	CUARP-100
G0	1	1	1	1	2
G1	3	6	6	6	15
G2	4	6	6	6	8
G3	14	17	17	17	27
G4	10	13	13	13	25
G5	11	14	14	14	30
G6	23	28	28	34	47
G7	13	15	17	17	28
G8	14	14	15	19	31

3.7 Conclusions

In this chapter, we have introduced two variants of the Collaboration Uncapacitated Arc Routing Problem. This is a profitable arc routing problem with multiple depots, where carriers may collaborate to improve the profit gained. An integer linear programming formulation with binary and integer variables has been presented, as well as a branch-and-cut algorithm. In the CUARP the goal is the maximization of the total profit of the coalition. The profit gained by the coalition of carriers never decreases with respect to the case without collaboration. Individual carriers, however, may lose profit in the collaboration scheme modeled by the CUARP. For this reason, we considered a variant where carriers may set thresholds on the profit gained in the collaborative scheme. This variant allocates the profit gained by the coalition in a more balanced way. Interestingly, we analyzed the cooperative game associated with the CUARP and noted that in the case of two carriers the t-CUARP profit allocation is stable because it belongs to the core of the game. Moreover, collaboration may produce an arbitrary large increase of the profit. A set of benchmark instances were generated and the results of extensive computational experiments presented and analyzed.

For future research, attention should be focused on the extension of the models proposed to include real-life features of the problem such as time and capacity constraints. Another direction concerns the design of heuristic algorithms for the solution of large instances.

4. METAHEURISTICS FOR THE COLLABORATION UNCAPACITATED ARC ROUTING PROBLEM

4.1 Introduction

In this chapter, we present an heuristic framework to solve large instances of the CUARP. This problem was defined in the previous chapter, it is part of Arc Routing Problems (ARPs) family with profit. Next, we briefly summarize the CUARP. The CUARP is arc routing problem which deals with situations where collaboration is managed in a centralized way. We consider a set of carriers cooperating under the guidance of a central station that acts in a non-partisan way. Each carrier has a depot and a set of customers, whose service generates a revenue. Each carrier identifies a subset of customers that it wants or needs to serve. The remaining customers are defined as shared customers. A shared customer may be served by the carrier that decided to share it, by a different carrier or not to be served. Part of the revenue of a shared customer that is served goes to the carrier that decided to share the customer and part goes to the carrier that actually serves it. If a shared customer is not served by any carrier the revenue is not collected and a penalty is paid. We assume that each carrier has one vehicle and that vehicle capacity is not relevant, that is the vehicles are uncapacitated. This yields to the study of an uncapacitated arc routing problem with multiple depots, where carriers collaborate to improve the profit gained. We called it Collaboration Uncapacitated Arc Routing Problem (CUARP) as stated in the previous chapter.

The chapter is structured as follows. In Section 4.2 we accurately describe the heuristic framework proposed to solve large CUARP instances. initial feasible solution for the CUARP. Section 4.3 presents the heuristic algorithm used to provide an initial feasible solution. Sections 4.4 and 4.5 describe neighborhood generation and the destroy and repair core of the heuristic framework. In particular, in Section 4.5 we outline two different implementation of such framework: an Variable Neighborhood Search and an Adaptive Large Neighborhood Search In Section 4.6 we describe the computational experiments, present the obtained results and compare the two heuristic algorithms. Finally, conclusions and remarks are given in Section 4.7.

4.2 The destroy and repair heuristic framework

In this section we present the destroy and repair heuristic framework we use to deal with CUARP instances. It is based on a cycle in which we repeat destroying and repairing the current solution with the aims of improving it. Indeed, a part from the initial solution, which is the starting point of the framework, the core of the destroy and repair approach is made up by neighborhoods generation and search and acceptance and stopping criteria used during the implementation. Next, we outline the structure of each possible implementation of this framework for solving the CUARP.

(First Step) Find an initial feasible solution by means of an adaption of famous Frederickson heuristic (*e.g.* Frederickson in [41]);

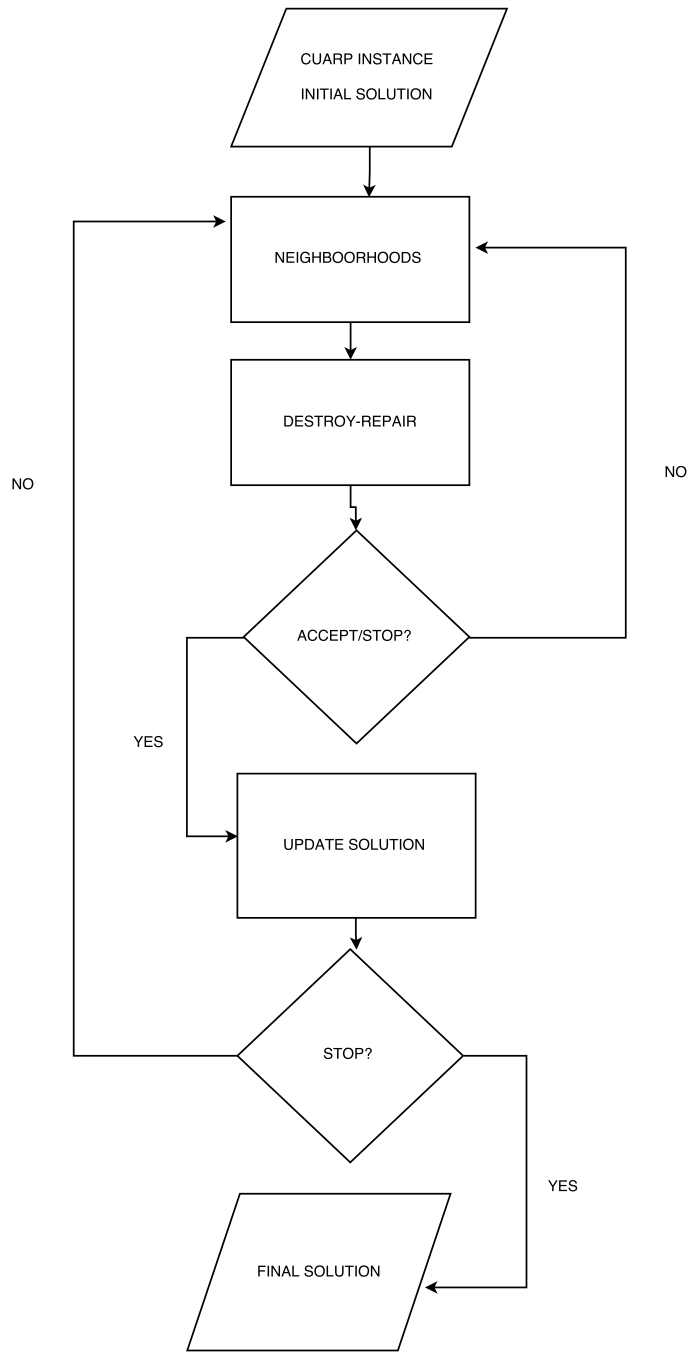
(Second Step) create a large set of neighborhoods;

(Third Step) use a ruin and repair approach:

- pick up one neighborhood by means of a local search engine;
- destroy the current solution (*i.e.* destroy the chosen neighborhood);
- repair the destroyed solution by means of the exact algorithm developed in Chapter 3;

(Fourth Step) back to the Second step until stopping criteria are met.

In Figure 4.2 we give a flowchart of the destroy and repair heuristic framework. In the next sections we analyze more deeply the main steps of these heuristic framework.


CUARP INSTANCE
INITIAL SOLUTION
NEIGHBOORHOODS
DESTROY-REPAIR
NO
NO
ACCEPT/STOP?
YES
UPDATE SOLUTION
STOP?
YES
FINAL SOLUTION

4.3 The initial solution

In this section, we present a general framework to generate an initial feasible solution z_I for the CUARP. An overview of the algorithm is given in Algorithm 4.1 and is mainly based on a directed version of the well known Frederickson heuristic (see Frederickson [41]) used for solving the Directed Rural Postman Problem (DRPP). In general, this initial solution can be used in the initialization phase of different heuristic, metaheuristic and matheuristic procedures developed to find a best CUARP solution.

Below, we provide a sketch of the initialization algorithm.

Algorithm 4.1: Initial Solution Procedure

Input: a $\mathcal{I}$ CUARP instance.

Output: an initial feasible solution z_I.

1 Apply the Preprocess procedure to $\mathcal{I}$.

2 Solve a DRPP for each depot using a directed variant of the Frederickson heuristic:

- Construct a minimum cost spanning arborescence on connected components.
- Solve a transportation problem.
- Determine an Eulerian Tour.
- Refine the Tour.

Put together each DRPP tour to form the initial feasible solution.

Hence, the search for an initial feasible solution rely on solving a DRPP for each carrier. Therefore, we have to define for each carrier the arcs that it has to traverse. We describe it in the preprocess procedure.

4.3.1 Preprocess procedure

In this subsection we describe a preprocess procedure which has to be done in order to simplify each of the next steps required to compute a feasible solution. Given a CUARP instance with original graph $G = (V, A)$, a defined number k of carriers and for each carrier l a shared S^l and a required R^l sets of customers. Let V_R^l and V_S^l sets such that they contain the vertices of all required and shared arcs by carrier l, respectively. We define for each carrier a set of arcs B^l that must be traversed in the initial solution and we identify B with $\cup B^l$. Hence, in order to produce a feasible solution we have to assign to each carrier at least all its own required arcs (*i.e.* we assign to carrier l at least the required set R^l). However, if we assign to a carrier only its required arcs the solution is very far from the optimal one. Hence, we decide to assign all the required arcs and a given percentage (0.5) of shared arcs to their respective carriers, to produce a better initial feasible solution. Next, after this assignment phase, we do some manipulation on the instance graph in order to simplify the computational strength of the following steps. Let V_B^l be the set of vertices of arcs which belong to B^l for carrier l. Now, we can define a G-subgraph G_B^l such as $G_B^l = (V_B^l, B^l)$ Indeed, we can solve a DRPP for each carrier on a simplified graph $G_S^l = (V_F^l, F^l)$, with $V_F^l = V_B^l$. Firstly, we include in F^l each arc belonging to B^l and each arc whose vertices are contained in V_R^l. Then we add to F^l all arcs (u, v) such that $u, v \in V_R^l$. The cost of such an added arc is equal to the shortest path linking u and v in G. Finally, we reduce the set F^l by eliminating:

(a) all arcs $(u, v) \in F$ such that $c_{uv} = c_{uk} + c_{kv}$ for some $k \in V_B$, and

(b) one of two parallel arcs if they have the same cost and the same orientation.

If the depot v^l does not belong to V_B^l than we include in F^l the arcs that cost less which contain the depot and go from and to a vertex in V_F^l.

At this point, for each depot l we perform a directed variant of Frederickson heuristic (see Frederickson [41]) in order to solve a DRPP.

4.3.2 Minimum arborescence step

In this subsection, we describe the Minimum arborescence step. Set $G_S^l = (V_F^l, F^l)$. Let $C_1, \ldots, C_c$ be the connected components of G_R. If there is only one connected components we go to the Transportation phase. Otherwise, we build up a complete directed new graph G' with c vertices such as arc $a = (p, q) \in G'$ cost is given by $\min\{c_{uv} | u \in C_u, v \in C_v\}$. Determine a minimum spanning arborescence T in G' rooted at any arbitrary vertex by means of Edmonds-Chu Liu algorithm as described in Edmonds [35]. For each arc in T consider its actual vertices in G and add to G^l_S all the arcs that belongs to the shortest path linking them in G.

This phase has a major drawback because a minimum cost spanning arborescence can heavily depend on its root. One solution is to repeat the above algorithm by considering in sequence all possible root, and then selecting the best solution. However, this procedure has very high computational costs and time. Thus, it has not been chosen, because our purpose is to find the best possible feasible solution in an acceptable period of time.

4.3.3 Transportation phase

In this subsection, we present the transportation phase of the Frederickson heuristic. The so-called Transportation phase can divided in three steps.

Step 1 Compute $s_u = d_u^+ - d_u^-$ for each vertex u in G^l_S, where d_u^+ and d_u^- are the indegree and the outdegree of vertex u in the graph G^l_S, respectively. In a directed graph, the indegree of a vertex is the number of arcs entering the vertex, while outdegree is the number of arcs exiting it. Let $I = \{u \in V | s_u > 0\}$ and $J = \{u \in V | s_u < 0\}$ be the sets of unbalanced vertices with more arcs entering and exiting, respectively.

Step 2 Solve the following transportation problem.

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\sum_{j \in J} x_{ij} = b_j$$

$$\sum_{i \in I} x_{ij} = b_i$$

$$x_{ij} > 0$$

where c_{uv} is the cost of the shortest path linking u and v in G^l_S.

Step 3 Let f_{uv} be the flow on arc (u, v) in the solution of the previous problem. Add to G^l_S f_{uv} copies of arcs that belong to the shortest path linking vertices u and v in G.

The resulting graph is a strongly connected directed multigraph with all vertices balanced which means that indegree and outdegree are equal for each vertex.

4.3.4 Eulerian tour step

In this step we determine an Eulerian tour E in G^l_S, which is a tour that covers all the arcs of graph G^l_S. Such a tour exists since the previous phases give us as a result a strongly connected multigraph such that at least one eulerian tour exists. This tour E is also a feasible tour for the directed rural postman problem with depot v^l and required arcs B^l.

4.3.5 Refinement step

This whole procedure can be improve by means of some refinement steps. Firstly, an Eulerian tour E can be improve by replacing two consecutive non-required arcs (u, v) and (p, q) on the tour, by an arc (u, q) provided that $(u, q) \in G$ and that the profit collected increases. Another refinement consists in locating paths $u - v - p - q - t$ formed by four arcs such that the arc (v, p) is required and while the other are not. These paths can be substituted if profitable by paths like $u - q - v - p - t$ provided that all arcs in the path belong to G_S. These two move can be repeated over and over, alternating them, until there is no improvement.

Finally, all the tours, which has been found in this way, are putting together to form the initial CUARP feasible solution.

Throughout the rest of this chapter, the initial solution is called z_I and is used as a starting point to implement the ruin and repair approach. In particular we propose two different implementation of our heuristic framework a Variable Neighborhood Search (VNS) and an Adaptive Large Neighborhood Search (ALNS).

4.4 Neighborhoods

Next, we describe how we create a large set of neighborhoods in order to run a proper search as previously stated. Firstly, a neighborhood is made up of paths of arcs which belong to the current solution. It may contain one or more paths. Indeed, we can discriminate among neighborhoods on the basis of size. For instance, if a neighborhood contains q paths than it is called a q-size neighborhood.

To better understand how works the neighborhood destruction and repair, we have to explain how neighborhoods are formed. First of all, we have to find suitable paths to destroy in the current solution. In accordance with the problem we are dealing with, the CUARP, we choose paths formed by shared and non-demand arcs. Once all paths are detected, we list them in five ways, according to their overall profitability (in descending or ascending order), to their length (in descending or ascending order) and randomly. In case there are only two or less paths we randomly break them such that finally we have at least five paths. Hence, secondly we list paths in accordance with various criteria. At this point we have five lists of paths from which we can select to form different neighborhoods. Finally to create a neighborhood we have to pick paths from lists according to fixed rules. The simplest way is to pick them as they are listed, which means if a q-neighborhood is needed we select the first q paths in the list. Another way is to pick them starting from the q-th path and then jumping to $q+2$-th, $q+4$-th, $\ldots$ until the neighborhood is completely formed. Obviously, if the paths listed are finished then the available paths are added until the neighborhood is complete (*i.e.* it reach its size q). These last neighborhoods can be generalized using a k-jump. For our purposes, we use a 2 and a 3 length jumps. Hence, at each iteration we have 15 different types of neighborhoods of different size from which we can choose to destroy the current solution.

4.5 The destroy and repair cycle

The destroy and repair cycle represents the core of the whole heuristic framework. Indeed, by destroying feasible solution we are able to search the neighborhoods that we create so that we can repair the destroyed solution and be able to find a better feasible solution. In particular, the repair phase of the cycle is carried out by means of the exact algorithm developed in Chapter 3. Moreover, in order to build an efficient destroy and repair cycle we have to define stopping and acceptance criteria.

4.5.1 Stopping Criteria

Algorithm 4.2: Stopping Criterion

Input : Route S

Stopping Criteria:
if *time over TimeMAX* **then**
    Stop
if *number of iterations over ItMAX* **then**
    Stop
if *no improvement in L iterations* **then**
    Stop
if *no more paths to removed* **then**
    Stop
end

Effective stopping criteria have to be defined in order to avoid to interrupt the heuristic framework too early or too late. Hence we propose a bunch of criteria. We stop the general heuristic if the time spent is gone over a threshold TimeMax (equal to two hours in our implementations) or if the number of iterations has exceeded a maximum number ItMAX which is set in a dynamic way as the maximum number of paths that can be removed from the solution which is currently handle by the algorithm. Moreover, we decide to stop the heuristic framework if after L (10) iterations there is no improvement at all.

At this point, we develope two variants changing the way destroy the solution and search the neighborhoods. Indeed we develop two metaheuristics: a Variable Neighborhood Search (VNS) and an Adaptive Large Neighborhood Search (ALNS). However, both metaheuristics share the same stopping criteria. In the following subsections we describe the VNS and the ALNS implementations of the destroy and repair heuristic framework. This two implementations differ because of different acceptance criteria and different way for searching the neighborhoods.

4.5.2 The Variable Neighborhood Search

In this section we describe the VNS procedure we use to solve the CUARP. VNS has been firstly proposed by Mladenovic et al. in [63] to solve Traveling Salesman Problem with or without backhauls. Principles and applications of the VNS are describe in Hansen et al. [47]. It has also been applied to more general Vehicle Routing Problem such as in Braysy et al. [18] and in Polacek et al. [70]. In general, VNS is a general local search framework that perform the search of neighborhoods until it finds the local minimum and then escapes from it by means of a perturbation or a destruction of the solution found. We perform a search of neighborhoods that can be destroy and then repair it using the exact formulation. Algorithm 4.3 provide a sketch of the VNS procedure we implement to solve the CUARP instances.

The VNS starts from an initial solution and improves it using the destroy and repair heuristic approach. Indeed, at first the initial solution is destroyed by removing one randomly chosen 1-size neighborhood as described in the algorithm 4.3. After removing this neighborhood the rest of the solution is fixed and it is given to the exact algorithm develop in Chapter 3. The exact algorithm repair the solution. Then, if the new solution is better then the current we repeat the removal and fixing procedure as described for the initial solution; otherwise we go deeper in our neighborhood removal enlarging the size of the neighborhood until the solution improves. When we find an improving solution we restart the whole procedure by removing a randomly chosen 1-size neighborhood until stopping criteria are met.

4.5.3 The Adaptive Large Neighborhood Search

In this section we describe the Adaptive Large Neighborhood Search (ALNS) procedure we apply to solve the CUARP. Such metaheuristic has been recently proposed by Ropke et al. in [72] and by Pisinger et al. in [69] to solve

Algorithm 4.3: Variable Neighborhood Search

Input : CUARP instance

Output: CUARP feasible solution

```
Construct an initial feasible solution z_I by means of heuristic
Algorithm 4.1;
Set current and best solution z_H = z_I, z_best = z_I;
Set n = 1; Create neighborhoods;
Choose randomly a type of neighborhood;
while Stopping Criterion is not met do
    Remove an n-size neighborhood of the chosen type;
    Generate a new solution z̃ from z repairing it with an exact
    algorithm;
    if z̃ is the best solution then
        Set z_best = z̃ and set z_H = z̃;
        Set n=1;
        Create new neighborhoods and choose one type randomly;
    end
    else
        Set n = n + 1
    end
end
Return z_best.
```

various vehicle routing problem, developed starting from the Large Neighborhood Search (LNS) provided by Shaw in [76]. Since than it has been applied to different contexts (see Laporte et al. in [55]). Hence, this algorithm has been shown to be flexible and highly suitable to solve complex problems and large instances. Indeed, it can handle various families of hard constrains easily because of an highly diversified search over multiple large neighborhoods and through the use of a random and adaptive local search engine. Algorithm 4.4 provide a sketch of the ALNS procedure we implement to solve the CUARP instances.

The ALNS differs from the VNS because it uses a roulette wheel selection algorithm to choose which neighborhood has to be destroyed. Moreover, a solution can be accepted even if it does not improve the current solution. A solution is accepted using a deterministic annealing algorithm which means that it is always accepted if it improves the current solution, otherwise it is accepted with a fixed probability. We set this probability of accepting a non-improving solution to 25.0%. Next, we describe the neighborhood selection procedure and the adaptive local search engine that help us in the random choice of the neighborhood taking into account past achievement of neighborhoods of the same kind.

Neighborhoods selection

When it comes to the choice of a neighborhood to be destroy at a given iteration the ALNS relies on a roulette-wheel procedure in which each type of neighborhood is assigned a weight depending on its past performance. Let π_i measures how well i-type neighborhoods have performed in the past. Hence, given h different type of neighborhoods with weights π_i, an j-type neighborhood will be selected with probability $p_j = \frac{\pi_j}{\sum_{i=1}^{h} \pi_i}$. Initially, all h probabilities p_i are equal to $\frac{1}{h}$. Each Q iteration, weights π_i are updated as follows:

$$\pi = \pi_{previousStep}(1 - r) + r(\frac{g_i}{\theta_i}); \tag{4.1}$$

where r is a reaction factor set to value comprised between 0 and 1, θ_i the number of times i-type of neighborhood is used and g_i are the scores initially

Algorithm 4.4: Adaptive Large Neighborhood Search

Input : CUARP instance

Output: CUARP feasible solution

Construct an initial feasible solution z_I *by means of heuristic Algorithm 4.1*;
Set current and best solution $z_H = z_I$, $z_{best} = z_I$;
Set $n = 1$; *Create neighborhoods*;
Set a probability for neighborhoods to be selected equal for the first step;
while *Stopping Criterion is* <u>*not*</u> *met* **do**
    Remove an n-size neighborhood, choosing a neighborhood by means of a roulette wheel selection;
    Generate a new solution $\tilde{z}$ *from* z *repairing it with an exact algorithm*;
    if $\tilde{z}$ *is the best solution* **then**
        Set $z_H = \tilde{z}$ and set $z_{best} = \tilde{z}$;
        Set $n = 1$;
    end
    if $\tilde{z}$ *met Acceptance Criterion (Deterministic Annealing)* **then**
        Set $z_H = \tilde{z}$
        Set $n = 1$;
    end
    else
        Set $n = n + 1$;
    end
    Update probability in accordance with the roulette wheel algorithm;
end
Return z_{best};

set to zero. Scores g_i are updated at each iteration by adding σ_1 if an i type of neighborhood gives the last best solution, σ_2 and σ_3 if the new solution reached using an i type of neighborhood is accepted and it improves or it does not improve the current solution, respectively. Obviously, $\sigma_3 < \sigma_2 < \sigma_1$. After updating weights π_i, scores are reset to zero.

This neighborhood selection procedure is the core of the ALNS. It relies on a random roulette-wheel procedure adjust through the use of an adaptive engine that needs to be tuned. Indeed, we tested different parameters combinations in a trial and error scheme. Finally we set the reaction factor r to 0.25, the parameter Q to 2, while we set σ parameters to $\sigma_1 = 30$, $\sigma_2 = 20$ and $\sigma_3 = 10$.

4.6 Experimental Analysis

In order to evaluate the performance of the proposed heuristic algorithm, we have run a series of computational experiments. We present in this section the obtained numerical results. Programs were coded in Java using CPLEX 12.5 library (64 bit) for the solution of the mixed integer problems. Default parameters were used. All tests were run on a HP Z400 Workstation, 64 bit, 3.33 GHz, 12.0 RAM.

4.6.1 Testing environment

We use the 118 CUARP benchmark instances developed in Chapter 3. Those instances are derived from the 118 PRPP benchmark instances used in [5]. We can divide them into 4 groups. The first two data sets are obtained from the Albaida Spain Graph (see Corberán and Sanchis [24]). The second group contains the 24 instances (labeled P) of Christofides et al. [19]. The last three groups contain instances from Hertz et al. [49]: 36 instances with vertices of degree 4 and RPP disconnected required edge sets (labeled D), 36 grid instances (labeled G), and 20 randomly generated instances (labeled R). A summary can be found in Chapter 3 in Table 3.3 which summarizes instances main characteristics.

In order to test our metaheuristics we also generate a set of large instances. We derive it from the UR Undirected Rural Postman Problem instances which can be found online at `http://www.uv.es/corberan/instancias.htm`. First we generate profits as follows:

- $b_e \in$ U$[c_e, 3c_e]$, if e is a required edge of the RPP instance;

where U[a,b] denotes the integer uniform distribution in the interval [a,b]; while b_e and c_e are the profit genrated and the original cost on edge e, respectively. Then, we transform each rural postman problem instances with profit as we have already done in the previous chapter to generate the CUARP benchmark instances. In particular, we set the number of depots to three. Table 4.1 summarizes information on these large instances. Columns under *#vertices* and *#arcs* give, respectively, the number of vertices and the number of arcs of the instance. Columns under $|R^1|$, $|R^2|$ and $|R^3|$ give the number of 1-required, 2-required arcs and 3-required arcs, respectively. Similarly, columns under $|S^1|$, $|S^2|$ and $|S^3|$ give the number of 1-shared, 2-shared arcs and 3-shared arcs. We note the number of vertices goes up to 1000 and that of the arcs to 995008. Hence, comparing them with the benchmark instances UR instances are large.

Tab. 4.1: Large instances characteristics

	vertices	*arcs*	$\lvert R^1\rvert$	$\lvert R^2\rvert$	$\lvert R^3\rvert$	$\lvert S^1\rvert$	$\lvert S^2\rvert$	$\lvert S^3\rvert$
UR132	605	363008	99	63	51	99	64	51
UR135	892	791212	226	120	113	227	120	114
UR137	980	955508	245	257	166	245	258	167
UR145	929	858404	321	89	143	322	89	144
UR147	996	987044	479	163	221	479	163	222
UR152	766	582934	136	115	87	136	116	87
UR155	975	945758	372	144	180	372	145	180
UR157	1000	995008	502	272	267	502	272	268
UR162	802	639202	211	80	83	211	80	84
UR165	980	955508	473	128	194	474	128	195
UR167	1000	995008	402	455	322	403	456	322
UR532	298	87322	51	26	28	51	27	28
UR535	458	207482	150	20	68	151	20	68
UR537	493	240592	208	59	78	209	60	79
UR542	343	115942	78	23	35	79	23	35
UR545	476	224204	168	50	80	169	50	80
UR547	498	245522	205	123	103	206	124	103
UR552	388	148612	93	37	36	94	38	36
UR555	490	237658	140	122	80	140	122	80
UR557	498	245522	215	161	141	215	161	141
UR562	416	170984	88	61	57	88	61	58
UR565	496	243544	178	120	111	178	120	112
UR567	499	246514	290	149	153	290	149	154
UR732	452	202052	68	46	49	69	46	49
UR735	662	434942	163	94	81	163	94	81
UR737	744	549824	195	189	118	195	190	118
UR742	538	286762	106	58	58	107	59	59
UR745	713	504812	183	124	105	183	124	106
UR747	745	551308	263	219	163	264	219	164
UR752	580	333508	107	85	67	108	85	68
UR755	724	520564	228	166	121	229	166	121
UR757	748	555772	394	177	178	394	178	178
UR762	593	348692	145	52	74	146	52	75
UR765	741	545384	293	168	142	293	168	143
UR767	749	557264	467	196	233	468	196	233

4.6.2 Computational results on CUARP benchmark instances

The computational results are summarized in Tables 4.2 and 4.3. For each group of instances the first two columns report the average percentage optimality gap and the average CPU time, respectively. The optimality gap is computed as $(z - z_H)/z * 100$, where z is the optimal solution provided by the branch and cut algorithm and z_H is the best feasible solution found by the heuristic algorithm. Column three gives the number of paths destroyed and repaired in the instances. The next three columns give statistics. Columns # Gap=0%, # 0%<Gap<1%, # 1%≤Gap<3% report the number of instances on which the algorithm find the optimal solution and the number of times the optimality gap lies in between 0 and 1 and 1 and 3, respectively. Columns Worst Gap % and Max time show the worst optimality gap and the max CPU time spent for each set of instances.

Table 4.2 gives us the results about CUARP resolution with VNS. We notice the 101 instances out of 118 are solved to optimality. While, 11 present a gap less than 1% only 6 instances a gap that lies between 1% and 3%. In particular, the worst gap is 2.14% with an overall average of 0.14%. Moreover, the average CPU time spent is 3.80 seconds that is pushed up to 32.56 seconds in an instance of G100 group.

Tab. 4.2: VNS results for CUARP instances

	Average Gap	Average time	# Gap=0%	# 0%<Gap<1%	# 1%≤Gap<3%	Worst Gap	Max time
AA	0.50	21.30	0	1	0	0.50	21.30
AB	0.00	6.32	1	0	0	0.00	6.32
P	0.05	0.44	22	2	0	0.73	1.76
D16	0.18	0.06	8	0	1	1.59	0.09
D36	0.10	0.39	8	1	0	0.91	0.60
D64	0.05	1.72	8	1	0	0.43	2.73
D100	0.24	8.24	7	1	1	1.97	11.76
G16	0.00	0.10	9	0	0	0.00	0.23
G36	0.11	1.04	8	1	0	0.98	2.34
G64	0.19	4.66	8	0	1	1.70	10.31
G100	0.08	28.19	8	1	0	0.76	32.56
R20	0.10	0.09	4	1	0	0.48	0.12
R30	0.43	0.28	4	0	1	2.14	0.35
R40	0.25	0.57	3	2	0	0.79	0.82
R50	0.42	1.14	3	0	2	2.10	1.28
all	0.14	3.80	101	11	6	2.14	32.56

In Table 4.3 we provide a summary of the results of ALNS over CUARP instances. We notice that the average gap is equal to 0. However, 2 instances show a very smaller gap. Indeed, the worst overall gap is only 0.17%. Hence, we can say that almost all instances are solved to optimality and that only 2 out 118 show a gap that is negligible related to the size of those instances. The average CPU time spent is 7.86 seconds with a max of 157.98 seconds for an instance in G100 group.

Tab. 4.3: ALNS results for CUARP instances

	Average Gap	Average time	# Gap=0%	# 0%<Gap<1%	# 1%≤Gap<3%	Worst Gap	Max time
AA	0.01	46.14	0	1	0	0.01	46.14
AB	0.00	20.15	1	0	0	0.00	20.15
P	0.00	0.42	24	0	0	0.00	1.75
D16	0.00	0.04	9	0	0	0.00	0.07
D36	0.00	0.38	9	0	0	0.00	0.59
D64	0.00	2.83	9	0	0	0.00	5.28
D100	0.02	14.25	8	1	0	0.17	18.68
G16	0.00	0.09	9	0	0	0.00	0.21
G36	0.00	1.35	9	0	0	0.00	3.31
G64	0.00	8.45	9	0	0	0.00	13.03
G100	0.0	66.12	9	0	0	0.0	157.98
R20	0.00	0.08	5	0	0	0.00	0.11
R30	0.00	0.27	5	0	0	0.00	0.34
R40	0.00	0.55	5	0	0	0.00	0.81
R50	0.00	1.12	5	0	0	0.00	1.26
all	0.00	7.86	116	2	0	0.17	157.98

Comparing VNS and ALNS results, we note that ALNS perform better than VNS. Indeed, ALNS finds more optimal solutions compared to VNS. However, in general VNS is faster than ALNS. For instance, G100 group instances are solved by VNS with an average time of 28.19, while ALNS solve them in an average of 66.12 seconds. Hence, we can say that ALNS consume more time but is more accurate with respect to the VNS. Indeed, the ALNS can be seen as an improvement of the VNS by means of a deterministic annealing local search at the master level and an adaptive engine to allow the procedure learn and find their way taking into account past goals and feasible solution which has been reached. However, both VNS and ALNS results are satisfactory from our point of view.

4.6.3 Computational experiments on large instances

In order to evaluate the performance of the metaheuristics on large instances, we first solve the CUARP relaxation (*i.e.* removing binary and integrality constraints on variables in the CUARP formulation) for the 118 benchmark instances. Then, we compute the percentage gap between the relaxed solution and the exact solution (computed in the previous chapter) for each instance. Table 4.4 summarize the results of these experiments. Columns under *relax* and $CUARP$ give the average relaxed and optimal results for each group of the CUARP benchmark instances. The last column give the average percentage *gap* ($100|\frac{relax-CUARP}{relax}|$).

Tab. 4.4: relaxed CUARP results

	relax	*CUARP*	*gap*
AA	28253.0	24509.0	15.3
AB	19226.0	17911.0	7.3
P	430.0	392.9	15.0
D16	1586.8	1220.9	36.5
D36	2697.1	2460.6	9.6
D64	3799.1	3616.6	6.4
D100	5042.7	4873.2	4.4
G16	16.6	13.0	54.0
G36	55.9	51.0	9.7
G64	109.3	103.7	8.1
G100	186.9	170.8	13.3
R20	57158.0	53101.8	6.6
R30	76009.8	69230.2	10.5
R40	103006.6	90811.8	14.9
R50	107352.8	96350.4	10.6
		average	15.9
		max	300.0
		min	0.0

We notice that the average gap is 15.9 % which goes up to 300.0 % in one case. In particular, in small instances like those in group G16 we have a bigger gap comparing to the other ones. Indeed, in general gap ranges between 0.00% and about 16.00% without considering G16 group instances.

Table 4.5 summarizes the results of VNS and ALNS on the large instances. We compare them with the solutions of the relaxed version of the CUARP in order to evaluate the performances of the two metaheuristics. Hence, columns under *relax*, VNS and $ALNS$ give the relaxed , VNS and ALNS results, respectively. Columns under $gapVNS$ and $gapALNS$ give the gap between the relaxed results and VNS and ALNS for each instances, respectively. The last two columns give the CPU time spent by the VNS and the ALNS, respectively.

We note that the gap for the VNS ranges between 0.55 and 74.48 with an average of 21.38. While, the average ALNS gap is 18.61 ranging between 0.55 and 74.37. Comparisons among these results and results given in Table

Tab. 4.5: Metaheuristics results on large instances

	$relax$	VNS	$ALNS$	$gapVNS$	$gapALNS$	$timeVNS$	$timeALNS$
UR132	36837.80	31210.45	34317.58	15.28	6.84	41.03	710.87
UR135	77668.09	71965.81	72076.63	7.34	7.20	123.24	1667.08
UR137	116250.77	106722.02	106873.00	8.20	8.07	187.20	1860.16
UR145	103217.62	77022.24	83402.15	25.38	19.20	107.33	1766.87
UR147	165851.00	127517.51	138987.67	23.11	16.20	124.18	1617.68
UR152	70174.98	61120.14	61123.00	12.90	12.90	76.91	780.88
UR155	147675.64	93702.82	120653.80	36.55	18.30	167.08	1607.71
UR157	219689.88	145350.37	189378.30	33.84	13.80	244.45	6777.68
UR162	82089.54	78122.38	78122.38	4.83	4.83	122.30	1660.78
UR165	179281.43	133180.49	135827.67	25.71	24.24	119.96	1186.78
UR167	262557.10	241090.17	241090.17	8.18	8.18	123.40	6668.68
UR532	25372.28	20673.71	20897.47	18.52	17.64	8.89	88.61
UR535	57781.51	42965.74	45007.31	25.64	22.11	27.14	717.72
UR537	79640.11	61057.42	62258.93	23.33	21.82	32.29	668.66
UR542	37908.05	33917.73	33917.73	10.53	10.53	12.95	687.81
UR545	82106.90	71960.97	71960.97	12.36	12.36	28.86	886.02
UR547	116002.60	95531.55	95531.55	17.65	17.65	26.83	686.62
UR552	48516.30	38467.62	40677.28	20.71	16.16	15.91	178.61
UR555	100007.17	74291.04	76378.30	25.71	23.63	31.36	617.66
UR557	153951.95	111012.97	113846.27	27.89	26.05	25.74	677.02
UR562	70210.85	64373.09	64373.09	8.31	8.31	20.44	607.62
UR565	130740.24	126705.05	126705.05	3.09	3.09	27.61	761.62
UR567	190843.10	189185.66	189185.66	0.87	0.87	32.76	667.06
UR732	32375.81	30817.50	30817.50	4.81	4.81	23.40	667.02
UR735	65795.21	61848.67	61848.67	6.00	6.00	55.69	768.67
UR737	98644.82	98101.02	98101.02	0.55	0.55	70.20	760.08
UR742	50898.49	12989.22	13047.13	74.48	74.37	30.58	677.62
UR745	86062.00	25201.42	29013.36	70.72	66.29	103.58	1678.77
UR747	143336.66	71184.90	73129.60	50.34	48.98	92.98	867.68
UR752	61115.70	27676.17	30759.25	54.72	49.67	54.91	781.67
UR755	121954.64	115211.49	115211.49	5.53	5.53	103.27	1067.67
UR757	180110.92	137334.58	137334.58	23.75	23.75	77.06	706.78
UR762	69598.20	51935.02	52902.14	25.38	23.99	48.05	807.86
UR765	149540.04	114976.54	127846.30	23.11	14.51	72.54	677.08
UR767	231368.58	201514.57	201514.57	12.90	12.90	82.84	686.68
			average	21.38	18.61	72.66	1248.51
			max	74.48	74.37	244.45	6777.68
			min	0.55	0.55	8.89	88.61

4.4 show that the average gap between relaxed and optimal solutions and the average gap between relaxed and metaheuristics solutions are quite comparable. Hence, VNS and ALNS results are satisfactory. Moreover, comparing CPU time spent by VNS and ALNS we note that the former is faster than the latter but the ALNS performs slightly better on some instances.

4.7 Conclusion

In this chapter we have presented two heuristic algorithms for solving the Collaboration Uncapacitated Arc Routing Problem that was introduced in the previous chapter. We have proposed two metaheuristic scheme: an ALNS and a VNS to deal with the CUARP. These methods make use of an elaborated version of the destroy and repair paradigm. In order to evaluate the performance of the algorithm we have compared the exact results obtained in Chapter 3 with that obtained by means of the heuristics described in this chapter. The VNS and the ALNS heuristic consume less time compared to the exact method we previously developed. These heuristic procedures do not always find the optimal solution. Nevertheless, as pointed out in the previous section the gap is very small in particular in relation to the ALNS heuristic algorithm. In conclusion, the results obtained are very satisfactory.

Future work may be devoted to the extension of the problem and its solution methods to the case of capacitated carriers, carriers with multiple vehicles and with time and drivers constraints.

5. CONCLUSIONS

In this thesis, we have introduced, modeled and solved a new arc routing problem, and called it CUARP, to deal with collaboration issues which arise in road transportation among carriers, shippers and other logistic providers. In particular, we took into account profit allocation problems and their impact on collaboration and profit making. Indeed, we formulated two different variants of the CUARP to allocate in different ways the profit made by the collaborating partners. We have developed exact and heuristic algorithms for their solution. In the next paragraphs we outline our main findings as well as suggestions for future research.

We have proposed a comprehensive overview of the existing works about horizontal collaboration in a road transportation environment in Chapter 2. We classify into various streams. In particular, we analyzed the literature dividing it into what are actually shared: customers and/or logistic assets such as vehicles and capacities. However, even if problems related to collaboration are quite interesting and have attracted a lot of attention from the Operations Research community, the literature is quite scarce. Finally, we evaluated the use of cooperative game theory tools and well-known allocation schemes during the phase of profit allocation.

We have introduced a new arc routing problem the Collaboration Uncapacitated Arc Routing Problem (CUARP) in Chapter 3. This problem deals with the optimization of a collaboration scheme among carriers. We focused on situations where collaboration is managed in a centralized way. We considered a set of carriers cooperating under the guidance of a central station that acts in a non-partisan way. Each carrier has a depot and a set of customers, whose service generates a revenue. Each carrier identifies a subset of customers that it wants or needs to serve. The remaining customers are defined as shared customers. A shared customer may be served by the carrier that decided to share it, by a different carrier or not to be served.

Part of the revenue of a shared customer that is served goes to the carrier that decided to share the customer and part goes to the carrier that actually serves it. If a shared customer is not served by any carrier the revenue is not collected and a penalty is paid. We assumed that each carrier has one vehicle and that vehicle capacity is not relevant, that is the vehicles are uncapacitated. This yielded to the study the CUARP, an uncapacitated arc routing problem with multiple depots, where carriers collaborate to improve the profit gained. We studied two variants of the CUARP. In the first one the goal is the maximization of the total profit of the coalition of carriers, independently of the individual profit of each carrier. The second variant includes a lower bound on the individual profit of each carrier. This lower bound may represent the profit of the carrier in the case no collaboration is implemented. We formulated mixed integer programming models for the two variants of the problem and study their properties and their relations with well-known arc routing problems. We solved the formulations for the two proposed variants with a branch-and-cut algorithm and quantify the impact of collaboration for a large set of benchmark instances.

Finally, in Chapter 4 we presented a destroy and repair heuristic framework for solving the CUARP. In particular, we developed two different metaheuristics: a Variable Neighborhood Search (VNS) and an Adaptive Large Neighborhood Search (ALNS). Both were tested on the same benchmark instances solved with the exact algorithm proposed in the previous chapter. Comparisons among metaheuristics and exact methods results showed the great effectiveness and efficiency of both the VNS and the ALNS. Moreover, we generated a set of large instances and found for each of them a feasible solution by means of the developed metaheuristics. The results showed to be satisfactory. Indeed, comparing the gap between the best feasible solutions found and the optimal relaxed solutions for large and benchmark instances, we found that the average gap was similar.

In this thesis we acknowledged the growing importance of horizontal collaboration in road transportation. Indeed, we examined different models and techniques that had already been used to investigate specific or general problem related to horizontal collaboration in road transportation. We studied profit, benefit and cost allocation systems beyond the mere proportional allocation. We investigated the impact and the effectiveness of collaboration on profit collecting, formulating a new arc routing problem, the CUARP, to address a particular case of carriers collaboration that has not yet been studied We solved the CUARP with an exact method and designed heuristic solution methods to solve very large CUARP instances.

Future works can be devoted the developments of new models to address realistic problems and the drawing up of more general and rich mathematical formulations to take into account various aspects of horizontal roadside collaboration. For instance, we should focused on the inclusion of real life features in the models such as time and capacity constraints. Since the main goal of different participants in a collaborative scheme is making profit, it can be interesting to refine old and develop new efficient profit or cost allocation procedures through the use of cooperative game theory tools and then the matching of these techniques with particular problems. It may also be stimulating to pursue an integration of more levels of supply chain in a collaborative perspective and strengthen vertical collaboration and integration among different levels and different problems in the supply chain. Finally, a promising research path to be follow is the use and study of specific profit and cost allocation techniques that put together operational research and game theoretical approaches to given collaboration problems.

Bibliography

[1] R. Agarwal and Ö. Ergun. Network design and allocation mechanisms for carrier alliances in liner shipping. *Operations Research*, 58(6):1726–1742, 2010.

[2] R. Agarwal, Ö. Ergun, L. Houghtalen, and O. Ö. Özener. Collaboration in cargo transportation. In *Optimization and Logistics Challenges in the Enterprise*, pages 373–409. Springer, 2009.

[3] D. Ahr. Contributions to Multiple Postmen Problems. PhD dissertation, Departament of Computer Science, Heidelberg University, Heidelberg, Germany. 2004.

[4] J. Aráoz, E. Fernández, and C. Franquesa. The clustered prize-collecting arc routing problem. *Transportation Science*, 43(3):287–300, 2009.

[5] J. Aráoz, E. Fernández, and O. Meza. Solving the prize-collecting rural postman problem. *European Journal of Operational Research*, 196(3):886–896, 2009.

[6] J. Aráoz, E. Fernández, and C. Zoltan. Privatized rural postman problems. *Computers & Operations Research*, 33(12):3432–3449, 2006.

[7] C. Archetti, G. Guastaroba, and M. Speranza. An ilp-refined tabu search for the directed profitable rural postman problem. *Discrete Applied Mathematics*, 163:3–16, 2014.

[8] C. Archetti and M. G. Speranza. Arc routing problems with profits. in *Arc Routing: Problems, Methods, and Applications.* Á. Corberán and G. Laprote (eds.) MOS-SIAM series on Optimization, 281–297, 2015.

[9] J.-F. Audy and S. D'Amours. Impact of benefit sharing among companies in the implantation of a collaborative transportation system-an

application in the furniture industry. In *Pervasive Collaborative Networks*, pages 519–532. Springer, 2008.

[10] J.-F. Audy, S. D'Amours, and M. Rönnqvist. An empirical study on coalition formation and cost/savings allocation. *International Journal of Production Economics*, 136(1):13–27, 2012.

[11] J.-F. Audy, S. D'Amours, and L.-M. Rousseau. Cost allocation in the establishment of a collaborative transportation agreement: an application in the furniture industry. *Journal of the Operational Research Society*, 62(6):960–970, 2011.

[12] J.-F. Audy, N. Lehoux, S. D'Amours, and M. Rönnqvist. A framework for an efficient implementation of logistics collaborations. *International transactions in operational research*, 19(5):633–657, 2012.

[13] E. Bailey, A. Unnikrishnan, and D.-Y. Lin. Models for minimizing backhaul costs through freight collaboration. *Transportation Research Record: Journal of the Transportation Research Board*, 2224(1):51–60, 2011.

[14] E. Ballot and F. Fontane. Reducing transportation co2 emissions through pooling of supply networks: perspectives from a case study in french retail chains. *Production Planning & Control*, 21(6):640–650, 2010.

[15] J. M. Belenguer and E. Benavent. The capacitated arc routing problem: Valid inequalities and facets. *Computational Optimization and Applications*, 10(2):165–187, 1998.

[16] E. Benavent, A. Corberán and J.M. Sanchis. Linear programming based methods for solving arc routing problems. In *Arc Routing: Theory, Solutions and Applications.* M. Dror (ed.) Kluwer Academic Publishers, 231–275, 2000.

[17] S. Berger and C. Bierwirth. Solutions to the request reassignment problem in collaborative carrier networks. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):627–638, 2010.

[18] O. Bräysy. A reactive variable neighborhood search for the vehicle-routing problem with time windows. *INFORMS Journal on Computing*, 15(4):347–368, 2003.

[19] N. Christofides, V. Campos, A. Corberán, and E. Mota. An algorithm for the rural postman problem on a directed graph. *Mathematical Programming Study*, 26: 155–166. Springer, 1986.

[20] C. Clifton, A. Iyer, R. Cho, W. Jiang, M. Kantarcioglu, and J. Vaidya. An approach to securely identifying beneficial collaboration in decentralized logistics systems. *Manufacturing & Service Operations Management*, 10(1):108–125, 2008.

[21] G. U. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.

[22] R. L. Clarke. An analysis of the international ocean shipping conference system. *Transportation journal*, pages 17–29, 1997.

[23] Á. Corberán, E. Fernández, C. Franquesa, and J. M. Sanchis. The windy clustered prize-collecting arc-routing problem. *Transportation Science*, 45(3):317–334, 2011.

[24] Á. Corberán and J. Sanchis. A polyhedral approach to the rural postman problem. *European Journal of Operational Research*, 79(1):95–114, 1994.

[25] F. Cruijssen, O. Bräysy, W. Dullaert, H. Fleuren, and M. Salomon. Joint route planning under varying market conditions. *International Journal of Physical Distribution & Logistics Management*, 37(4):287–304, 2007.

[26] F. Cruijssen, M. Cools, and W. Dullaert. Horizontal cooperation in logistics: opportunities and impediments. *Transportation Research Part E: Logistics and Transportation Review*, 43(2):129–142, 2007.

[27] F. Cruijssen, W. Dullaert, and H. Fleuren. Horizontal cooperation in transport and logistics: a literature review. *Transportation journal*, pages 22–39, 2007.

[28] S. Dahl and U. Derigs. Cooperative planning in express carrier networks - an empirical study on the effectiveness of a real-time decision support system. *Decision Support Systems*, 51(3):620–626, 2011.

[29] B. Dai and H. Chen. A multi-agent and auction-based framework and approach for carrier collaboration. *Logistics Research*, 3(2-3):101–120, 2011.

[30] B. Dai and H. Chen. Mathematical model and solution approach for carriers' collaborative transportation planning in less than truckload transportation. *International Journal of Advanced Operations Management*, 4(1):62–84, 2012.

[31] B. Dai and H. Chen. Profit allocation mechanisms for carrier collaboration in pickup and delivery service. *Computers & Industrial Engineering*, 62(2):633–643, 2012.

[32] U. Derigs and T. Döhmer. Router: A fast and flexible local search algorithm for a class of rich vehicle routing problems. In *Operations Research Proceedings 2004*, pages 144–149. Springer, 2005.

[33] Y. Dong and K. Xu. A supply chain model of vendor managed inventory. *Transportation Research Part E: Logistics and Transportation Review*, 38(2):75–95, 2002.

[34] T. S. Driessen. A survey of consistency properties in cooperative game theory. *SIAM review*, 33(1):43–59, 1991.

[35] J. Edmonds. Optimum branchings *Journal of Research of the National Bureau of Standards B*, 71(4): 233-240, 1967.

[36] Ö. Ergun, G. Kuyzu, and M. Savelsbergh. Reducing truckload transportation costs through collaboration. *Transportation Science*, 41(2):206–221, 2007.

[37] Ö. Ergun, G. Kuyzu, and M. Savelsbergh. Shipper collaboration. *Computers & Operations Research*, 34(6):1551–1560, 2007.

[38] T. Fan, L. Vigeant-Langlois, C. Geissler, B. Bosler, and J. Wilmking. Evolution of global airline strategic alliance and consolidation in the twenty-first century. *Journal of Air Transport Management*, 7(6):349–360, 2001.

[39] S. E. Fawcett, G. M. Magnan, and M. W. McCarter. Benefits, barriers, and bridges to effective supply chain management. *Supply Chain Management: An International Journal*, 13(1):35–48, 2008.

[40] M. A. Figliozzi. Analysis and evaluation of incentive-compatible dynamic mechanisms for carrier collaboration. *Transportation Research Record: Journal of the Transportation Research Board*, 1966(1):34–40, 2006.

[41] G. N. Frederickson. Approximation algorithms for some postman problems. *Journal of the ACM (JACM)*, 26(3):538–554, 1979.

[42] M. Frisk, M. Göthe-Lundgren, K. Jörnsten, and M. Rönnqvist. Cost allocation in collaborative forest transportation. *European Journal of Operational Research*, 205(2):448–458, 2010.

[43] B. S. Fugate, B. Davis-Sramek, and T. J. Goldsby. Operational collaboration between shippers and carriers in the transportation industry. *The International Journal of Logistics Management*, 20(3):425–447, 2009.

[44] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial & Applied Mathematics*, 9(4):551–570, 1961.

[45] M. Göthe-Lundgren, K. Jörnsten, and P. Värbrand. On the nucleolus of the basic vehicle routing game. *Mathematical Programming*, 72(1):83–100, 1996.

[46] D. Gusfield. Very simple methods for all pairs network flow analysis. *SIAM Journal Applied Mathematics* 19, 1, 143–555, 1990.

[47] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.

[48] S. Hernández and S. Peeta. Centralized time-dependent multiple-carrier collaboration problem for less-than-truckload carriers. *Transportation Research Record: Journal of the Transportation Research Board*, 2263(1):26–34, 2011.

[49] A. Hertz, G. Laporte, and P. N. Hugo. Improvement procedures for the undirected rural postman problem. *INFORMS Journal on Computing*, 11(1):53–62, 1999.

[50] J. Holmström, K. Främling, R. Kaipia, and J. Saranen. Collaborative planning forecasting and replenishment: new solutions needed for mass collaboration. *Supply Chain Management: An International Journal*, 7(3):136–145, 2002.

[51] L. Houghtalen, Ö. Ergun, and J. Sokol. Designing mechanisms for the management of carrier alliances. *Transportation Science*, 45(4):465–482, 2011.

[52] C. Kilger and B. Reuter. Collaborative planning. In *Supply Chain Management and Advanced Planning*, pages 223–237. Springer, 2002.

[53] M. A. Krajewska and H. Kopfer. Collaborating freight forwarding enterprises. *OR spectrum*, 28(3):301–317, 2006.

[54] M. A. Krajewska, H. Kopfer, G. Laporte, S. Ropke, and G. Zaccour. Horizontal cooperation among freight carriers: request allocation and profit sharing. *Journal of the Operational Research Society*, 59(11):1483–1491, 2008.

[55] G. Laporte, R. Musmanno, and F. Vocaturo. An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands. *Transportation Science*, 44(1):125–135, 2010

[56] J. K. Lenstra and A. H. G. Rinnooy Kan. On general routing problems. *Networks*, 6(3):273–280, 1976.

[57] P. Liu, Y. Wu, N. Xu, et al. Allocating collaborative profit in less-than-truckload carrier alliance. *Journal of Service science and Management*, 3(01):143, 2010.

[58] R. Liu, Z. Jiang, R. Y. Fung, F. Chen, and X. Liu. Two-phase heuristic algorithms for full truckloads multi-depot capacitated vehicle routing problem in carrier collaboration. *Computers & Operations Research*, 37(5):950–959, 2010.

[59] F.-H. Liu and S.-Y. Shen. The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research Society*, pages 721–732, 1999.

[60] S. Lozano, P. Moreno, B. Adenso-Díaz, and E. Algaba. Cooperative game theory approach to allocating benefits of horizontal cooperation. *European Journal of Operational Research*, 229(2):444–452, 2013.

[61] R. Mason, C. Lalwani, and R. Boughton. Combining vertical and horizontal collaboration for transport optimisation. *Supply Chain Management: An International Journal*, 12(3):187–199, 2007.

[62] M. D. Meyer, S. Campbell, D. Leach, and M. Coogan. Collaboration: The key to success in transportation. *Transportation Research Record: Journal of the Transportation Research Board*, 1924(1):153–162, 2005.

[63] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.

[64] T. H. Oum, J.-H. Park, and A. Zhang. *Globalization and strategic alliances: The case of the airline industry.* 2000.

[65] O. Ö. Özener and Ö. Ergun. Allocating costs in a collaborative transportation procurement network. *Transportation Science*, 42(2):146–165, 2008.

[66] O. Ö. Özener, Ö. Ergun, and M. Savelsbergh. Lane-exchange mechanisms for truckload carrier collaboration. *Transportation Science*, 45(1):1–17, 2011.

[67] S. Pan, E. Ballot, and F. Fontane. The reduction of greenhouse gas emissions from freight transport by pooling supply chains. *International Journal of Production Economics*, 143(1):86–94, 2013.

[68] J.-H. Park. The effects of airline alliances on markets and economic welfare. *Transportation Research Part E: Logistics and Transportation Review*, 33(3):181–195, 1997.

[69] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & operations research*, 34(8):2403–2435, 2007.

[70] M. Polacek, R. F. Hartl, K. Doerner, and M. Reimann. A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of heuristics*, 10(6):613–627, 2004.

[71] H. J. Quak. Improving urban freight transport sustainability by carriers–best practices from the Netherlands and the EU project citylog. *Procedia-Social and Behavioral Sciences*, 39:158–171, 2012.

[72] S. Ropke and D. Pisinger. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3):750–775, 2006.

[73] D. Schmeidler. The nucleolus of a characteristic function game. *SIAM Journal on applied mathematics*, 17(6):1163–1170, 1969.

[74] D. Seifert. *Collaborative planning, forecasting, and replenishment: how to create a supply chain advantage.* AMACOM Div American Mgmt Assn, 2003.

[75] L. S. Shapley. A value for n-person games. Technical report, DTIC Document, 1952.

[76] P. Shaw. A new local search algorithm providing high quality solutions to vehicle routing problems. *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK*, 1997.

[77] E. J. Sheppard and D. Seidman. Ocean shipping alliances: the wave of the future? *International Journal of Maritime Economics*, 3(4):351–367, 2001.

[78] R. J. Sherman. Collaborative planning, forecasting & replenishment: Realizing the promise of efficient consumer response through collaborative technology. *Journal of Marketing Theory and Practice*, pages 6–9, 1998.

[79] T. M. Simatupang and R. Sridharan. The collaborative supply chain. *International Journal of Logistics Management, The*, 13(1):15–30, 2002.

[80] G. Stefansson. Collaborative logistics management and the role of third-party service providers. *International Journal of Physical Distribution & Logistics Management*, 36(2):76–92, 2006.

[81] D. J. Thomas and P. M. Griffin. Coordinated supply chain management. *European journal of operational research*, 94(1):1–15, 1996.

[82] S. H. Tijs and T. S. Driessen. Game theory and cost allocation problems. *Management Science*, 32(8):1015–1028, 1986.

[83] L. Verdonck, A. Caris, K. Ramaekers, and G. K. Janssens. Collaborative logistics from the perspective of road transportation companies. *Transport Reviews*, 33(6):700–719, 2013.

[84] M. Waller, M. E. Johnson, and T. Davis. Vendor-managed inventory in the retail supply chain. *Journal of business logistics*, 20:183–204, 1999.

[85] X. Wang and H. Kopfer. Collaborative transportation planning of less-than-truckload freight. *OR spectrum*, 36(2):357–380, 2014.

[86] G. Zhou, Y. V. Hui, and L. Liang. Strategic alliance in freight consolidation. *Transportation Research Part E: Logistics and Transportation Review*, 47(1):18–29, 2011.

APPENDIX

A. ELECTRONIC APPENDIX