The Shared Customer Collaboration Vehicle Routing Problem

Elena Fernández[a,b], Mireia Roca-Riu[a,c,*], M. Grazia Speranza[d]

[a] *Barcelona Tech-UPC, Department of statistics and operations research*
[b] *Barcelona Graduate School of Mathematics (BGSMath)*
[c] *Swiss Federal Institute of Technology Zürich (ETHZ), Institute for Transport Planning and Systems, Stefano-Franscini-Platz 5, HIL F 34.2, Zürich, Switzerland*
[d] *Università di Brescia, Dip. di Metodi Quantitativi*

Abstract

This paper introduces a new vehicle routing problem that arises in an urban area where several carriers operate and some of their customers have demand of service for more than one carrier. The problem, called Shared Customer Collaboration Vehicle Routing Problem, aims at reducing the overall operational cost in a collaboration framework among the carriers for the service of the shared customers. Alternative mathematical programming formulations are proposed for the problem that are solved with a branch-and-cut algorithm. Computational experiments on different sets of benchmark instances are run to assess the effectiveness of the formulations. Moreover, in order to estimate the savings coming from the collaboration, the optimal solutions are compared with the solutions obtained when carriers work independently from each other.

Keywords: Vehicle routing problem, carriers collaboration, mixed integer programming, branch-and-cut algorithm, urban logistics

1. Introduction

Most major cities present a dense and complex urban fabric, which hinders considerably last-mile deliveries. Multiple carriers offer delivery services through the city, involving numerous simultaneous trips to common areas, consuming partial loads and emanating from different depots. As a consequence last-mile deliveries generate various negative effects, such as high carriers costs, high traffic and space occupancy, or pollution, all of which are highly inconvenient for citizens. Collaboration among carriers who must serve common customers within the same time period may result in significant savings in such an scenario. Carriers could serve part of the demand for other carriers without deteriorating their own routes, better exploiting the vehicles capacity, thus obtaining savings both in terms of number of vehicles used and distance travelled. This is precisely the focus of this paper, where we introduce the Shared Customer Collaboration Vehicle Routing Problem (SCC-VRP), a new collaboration model that optimizes the potential benefits derived from alliances among carriers in this setting.

The benefits of collaboration in the freight transportation sector have raised increasing attention, especially in the last decade due to the availability of communication technologies that enable

*Corresponding author
Email addresses: `e.fernandez@upc.edu` (Elena Fernández), `mireia.roca-riu@upc.edu` (Mireia Roca-Riu), `grazia.speranza@unibs.it` (M. Grazia Speranza)

collaboration. Collaboration among companies at the same level of the supply chain is known as horizontal cooperation [16, 39]. When dealing with road transportation, horizontal cooperation among carriers can be further classified according to the operational collaboration mode in order sharing and capacity sharing (see for instance the recent survey [45]). Order sharing includes all situations where collaborating carriers combine, share or exchange customers orders or requests. In this setting the fleet of each collaborating carrier remains unchanged, as well as its trucks, which remain located at the same depots, from which the carrier's order delivery routes are performed. Capacity sharing on the other hand includes scenarios where carriers may acquire capacity from collaborative partners to satisfy their customer demand. In this case collaborating carriers do not share customer requests and every carrier delivers its individual order set.

The SCC-VRP deals with horizontal collaboration with order sharing in the framework of last-mile deliveries. Multiple carriers jointly operate in the same area, each of them, serving its own customers, from its own depot with its own fleet of vehicles. While some customers require service exclusively from only one carrier, others have service demand for multiple carriers (*shared customers*). Broadly speaking, the objective of the SCC-VRP is to exploit the benefits derived from allowing carriers to deliver products to the shared customers on behalf of other carriers. One specific characteristics of the SCC-VRP is that different carriers operate from different depots. Another characteristic is that not all customers can be shared. Moreover, the subset of carriers that can serve a given shared customer is not fixed, as it depends on the customer. We are not aware of any work dealing with carriers collaboration where any of these characteristics has been addressed.

The literature on carriers collaboration is certainly very large and an extensive review of the state of the art for all collaboration modes is outside the scope of this paper. We thus refer the interested reader to the survey [45] for a comprehensive and detailed discussion of the related literature, and we briefly overview here the main contributions and recent works with a closer relation to the operational problem that we study. Few works address quantitative models for decision support to carriers in a collaborative framework. Some papers propose methods to optimize collaboration among carriers in line-haul environments, either using mixed integer programming models (see, for example, [1]) or proposing a simple estimating formula, as in [10]. Auction systems that model the interactions among partners in a collaboration framework have been studied in [11, 18, 23, 33]. A model where all collaborating carriers offer all their requests for exchange, which seems rather unrealistic in practice, is studied in [47].

A few other papers analyze instead the possibility for companies to exchange customers or to form alliances. In [9] a framework is proposed to build and manage inter-firm relationships in the logistics area based on three case studies. In the case of [17], on-line cooperative express networks are proposed. Dynamic pickup and delivery and a decision support system are studied. The formation of alliances is studied for the problem of liner shipping in [2].

Some authors [15, 20, 21, 30, 38, 44] have addressed the collaboration problem from a joint route planning perspective where the overall demand of each customer is totally served from one single carrier. In particular, [15] estimate the synergy from the combination of outsourcing and horizontal cooperation by modelling the problem as a single-depot vehicle routing problem with time windows. The lane covering problem, which tries to identify repeatable or dedicated truckload tours for companies that regularly send truckload shipments that traverse (*cover*) some links (*lanes*) of the distribution network, was first considered in [20, 21] and later in [38] from a multi-company perspective. Other approaches consider specific features like: the use of resource pooling based on

Erlang delay systems [30] or the partners' flexibility [44] in order to evaluate cooperation.

A major issue in an order sharing collaboration framework is how the savings are shared among the coalition of the collaborating carriers. The behavior of collaborating partners was modelled in [31, 32] with a game theoretic approach in three phases: preprocessing, profit optimization and profit sharing. Also in [35] the problem of allocating the joint cost savings of the cooperation is tackled using cooperative game theory.

One of the few papers dealing with routing aspects of collaboration is [46]. Their model integrates transshipment into the conventional pickup and delivery problem with collaboration. A Mixed Integer Programming (MIP) formulation is presented for a problem that allows exchanging requests between carriers and the possibility of demand exchange at pre-specified transshipment points. An arc routing problem is analyzed in [22] to model collaboration in truckload shipping. A lower bound on the individual profit of each carrier is set in the optimization model to guarantee that all carriers benefit from the collaboration. A collaborative version of a routing problem with profits is proposed in [19], where the customers of different companies form a coalition and all customers can be served with the joint available resources. The formulations incorporate different cost allocation rules that can guarantee the desired behavior of the participants. Again, the coalition considers that all customers can be shared in the system.

Similarly to some of the works referenced above, the SCC-VRP focuses on the routing aspects of collaboration. Due to its specific characteristics the SCC-VRP gives rise to a new vehicle routing problem. In fact, the SCC-VRP generalizes the classical Multiple Depot Vehicle Routing Problem (MDVRP) (see, for example, [41, 42]) and becomes the MDVRP when no customer is shared. On the one hand the vehicle routing problem underlying the SCC-VRP must be stated on a multi-depot setting, as carriers do not operate from a common origin (consolidation center) but from different garages/warehouses. On the other hand, deciding the allocation of customers to carriers becomes involved as shared customers may be visited by one or more carrier. Moreover, even if it were known that a customer would be served by more than one carrier, the amount of demand served by each of them would not be known in advance, as carriers could decide to *interchange* their served demands to better use of the capacity of their vehicles. This aspect somehow relates the SCC-VRP with split delivery routing problems in which the overall demand of customers can be split among the vehicles routes. The essential difference with respect to classical split-delivery models is that in the SCC-VRP the overall customers demands cannot be split arbitrarily, since only quantities corresponding to individual customer/carrier orders can be served in the different routes. This feature is indeed one of the main difficulties of the SCC-VRP for which setting suitable mathematical programming formulations becomes a challenge on its own.

Potential applications for the SCC-VRP include regular deliveries to bars and restaurants, daily parcel deliveries, or other deliveries of goods of various nature. As an example, [15] describes the real case of three Dutch companies of distribution of frozen products. The companies had a considerable amount of overlap between customers, on average 68%, and collaboration reduced the traveled distance by 30.8% and the fleet size by 50%. Nevertheless, in other initiatives mentioned in that work, the savings range from 15% to 30%. Another example can be found on [40], where a case from the German food industry is presented and several manufacturers with same customers but complementary food products share their vehicle fleets to deliver their customers. Results show that cooperative scenarios outperform the non-cooperative one.

The scope and contributions of this paper are two-fold:

- Modeling. We introduce a new model for horizontal collaboration with order sharing, with potential applications in the framework of last-mile deliveries, and we quantify the tradeoff derived from this type of collaboration with respect to the situation where carriers work independently from each other. For this, we give a theoretical bound on the savings that can be obtained and show that the bound is tight. We also perform an empirical analysis based on the results from computational experiments on several sets of benchmark instances. The results indicate that the average cost savings range from 10.7% to 18.2% on random instances and from 2.5% to 16.4% on clustered instances. We finally present an extension of the SCC-VRP that includes the cost of transferring goods between depots, which may be needed in some circumstances in a collaborative framework and compare the savings with the transferring cost.

- Methodological and algorithmic. We study the new vehicle routing problem that arises when the SCC-VRP is addressed from a mathematical programming perspective. For this, we propose and study two alternative mixed integer linear programming (MILP) formulations for the SCC-VRP. The first formulation is a *vehicle-based* formulation, which follows the spirit of classical formulations for the Multiple Depot Vehicle Routing Problem (MDVRP) [27], by associating decision variables with the vehicles routes, both for the arcs that are traversed and for the customers that are visited. Even if reinforced with several families of valid inequalities the *vehicle-based* formulation is computationally cumbersome. Thus, following the current trend in complex vehicle routing problems [7, 34], a *load-based* formulation is also proposed. The main advantage of this formulation is that the number of binary decision variables reduces notably, since they are only associated with depots, but no longer with vehicles. However, this comes at the expenses of an additional set of continuous load variables, which are needed to guarantee that the balance constraints *redistribute* correctly the loads of the different routes. For each formulation we discuss several families of valid inequalities as well as the solution to the separation problems for the families of constraints of exponential sizes. An exact branch-and-cut algorithm is proposed for the solution of each formulation. Computational experiments on different sets of benchmark instances compare the performance of the two proposed formulations and find the maximum size of instances that can be solved to optimality with the best formulation.

The remainder of this paper is structured as follows. In Section 2 the SCC-VRP is introduced and the theoretical bound on the maximum saving that can be obtained through collaboration is derived. The *vehicle-based* and the *load-based* formulations are presented in Section 3. Section 4 gathers the methodological aspects of the solution algorithms that we propose. In particular, for each formulation, several families of valid inequalities are proposed, and their corresponding separation algorithms presented. The section ends with the detailed description of the branch-and-cut algorithm for each case. Section 5 describes the computational experiments. We present the numerical results, compare the MILP formulations, and analyze the structure of the obtained solutions under different possible scenarios. We close the paper in Section 6 with some comments and promising lines for further research.

2. The Shared Customer Collaboration Vehicle Routing Problem

We consider a set of transport companies (carriers) that operate in the same urban area with a high customer density and are willing to collaborate to reduce distribution costs. We assume that some customers are *shared* by different carriers, in the sense that they have demand (request goods) from more than one carrier. One shared customer may represent a group of individual customers that may request service from different carriers but are located close enough that one carrier could serve them with one stop in the delivery route. In this context, collaboration among carriers means that each of them is willing to transfer a part of its demand to other carriers, namely the part of its demand corresponding to some of the shared customers. Customers will only be transferred when the transfer decreases the overall distribution cost. Below we give a formal definition of the Shared Customer Collaboration Vehicle Routing Problem (SCC-VRP).

Let C denote a given set of carriers operating in a given area and N the set of customers in the area. We will denote by $m = |C|$ and $n = |N|$ the number of carriers and customers, respectively. Each carrier $r \in C$ has its own depot o_r and a homogeneous fleet of vehicles of capacity Q (the same for all carriers). Let $G = (V, A)$ denote the complete directed network, where $V = N \cup (\cup_{r \in C}\{o_r\})$ is the set of customers plus the depots, and A is the set of arcs connecting each pair of customers and each customer with the depots, i.e., $A = V \times V$. Associated with each arc $(i, j) \in A$ there is a travel cost $c_{ij} \geq 0$. We assume that travel costs satisfy the triangle inequality.

For $i \in N$, $r \in C$, $d_i^r \geq 0$ denotes the demand of customer i with respect to carrier r. When $d_i^r > 0$ we say that *i is a customer for carrier r*. We denote by N_r the set of customers for carrier $r \in C$. For $i \in N$, the set of carriers that have i as customer is denoted by $C_i \subseteq C$ and referred to as the *set of carriers for customer i*. Indeed, for $i \in N$, $r \in C$, $i \in N_r$ if and only if $r \in C_i$. When $C_i = \{r\}$ for $i \in N$, i.e. $|C_i| = 1$, then the demand of customer i must be served by carrier r. On the contrary, if $|C_i| > 1$, the demand d_i^s of customer i for carrier $s \in C_i$ can be *transferred* to any carrier $r \in C_i$, meaning that it can be served by carrier r. Moreover, *interchanging* the demands of a customer between two of its carriers is allowed. That is, for a customer $i \in N$ and two carriers $r, s \in C_i$, it is possible that carrier s serves the demand d_i^r and carrier r serves the demand d_i^s. On the contrary, splitting the demand of a customer for a carrier among several carriers is not allowed. Hence, each service demand d_i^s, $i \in N$, $s \in C_i$ must be entirely served by the same carrier $r \in C_i$ (not necessarily carrier s).

In the SCC-VRP each carrier performs a set of routes, starting and ending at its depot. The overall demand served by each route cannot exceed the capacity Q. For each customer $i \in N$, each of its service demands d_i^r, $r \in C_i$ must be allocated to a route of some of its carriers C_i. The objective is to minimize the total cost of the routes of the carriers.

We note that, as the arc costs satisfy the triangle inequality, there always exists an optimal solution where any carrier $r \in C$ only visits a subset of N_r, that is, carrier r does not visit any customer $i \in N \setminus N_r$. Hence, in the routes associated with carrier r the only arcs that can be traversed are the ones in $A^r = \{(i, j) \in A : i, j \in N_r, \text{ or } (i = o_r \text{ and } j \in N_r), \text{ or } (i \in N_r \text{ and } j = o_r)\}$.

The SCC-VRP is NP-hard, as the particular case with one single carrier reduces to the well-known Vehicle Routing Problem (VRP) (see, for instance, [43]). Another well-known particular case of the SCC-VRP is the Multiple Depot Vehicle Routing Problem (MDVRP) (see, for instance, [12, 26, 27, 41, 42, 48]), which arises when all customers are shared, i.e. $C_i = C$ for all $i \in N$. However, as we note below, the SCC-VRP is more general than the MDVRP.

- When a shared customer is not assigned to one single carrier (depot), i.e. $|C_i| > 1$, then it can be served by each company separately, by only one of the companies, or by any other combination. Furthermore, when a shared customer is served by more than one company, the exact amount that will be served by each company has to be decided, since interchanging the demands of a customer between two of its carriers is allowed.

- The SCC-VRP also models the case when carriers may forbid that some of their customers are served by a different carrier. Suppose that (due to marketing reasons) carrier $r \in C$ wants to serve customer $i \in N_r$, with $|C_i| > 1$. This case can be easily modeled with the SCC-VRP by just defining a *copy* of customer i, say i', co-located with i, and with $C_{i'} = \{r\}$.

- The SCC-VRP also allows to model the reverse case, where several customers, each of them with demand for only one of the carriers, are located close enough so that one carrier could serve all of them with one stop in the delivery route. For this, all such customers should be *merged* into a single one with the same demand with respect to each of the carriers.

In fact, the SCC-VRP can be seen as a variant of a multi-depot split delivery VRP, in which the overall demand of each customer, $D_i = \sum_{r \in C} d_i^r$ can be *split* among the routes of the different carriers. The essential difference with respect to traditional split-delivery models is that D_i cannot be split arbitrarily. Only quantities corresponding to the individual demands d_i^r can be served in the different routes. Moreover each individual demand d_i^r must be served by one of the routes. This feature is indeed one of the main difficulties of the SCC-VRP as specific decision variables are needed in order to decide the carrier and specific route that will serve each demand d_i^r.

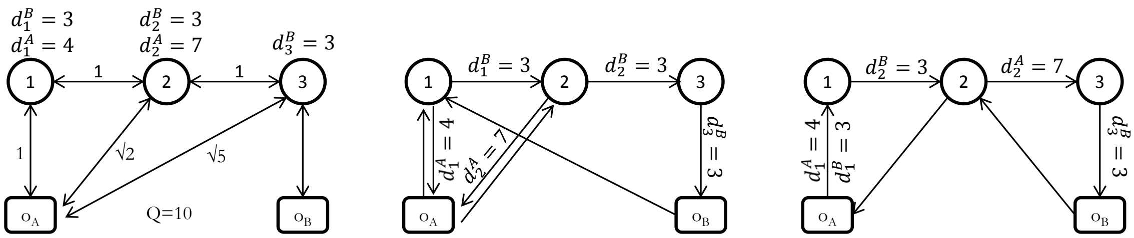
We illustrate the SCC-VRP with an example and then show how large the savings can be.

Example: Figure 1 shows an example of a SCC-VRP instance with 2 carriers, A and B, 3 customers and vehicle capacity $Q = 10$. For ease of presentation Figure 1(a) only depicts a few of the arcs of the complete network and their associated Euclidean costs, where each pair of horizontally or vertically consecutive nodes are at distance 1. In this example $N_A = \{1, 2\}$ and $N_B = \{1, 2, 3\}$, i.e., customers 1 and 2 have demand for both carriers, whereas customer 3 has demand for carrier B only. Figure 1(b) shows the optimal solution when no collaboration exists. Since $Q = 10$, carrier A needs two routes to serve its two customers, which, in total, have a demand of 11. In contrast, carrier B may serve its three customers in one single route. The overall cost of the solution without collaboration is $5+2\sqrt{2}+\sqrt{5}$. Figure 1(c) gives the optimal solution when collaboration is allowed. Now, the demand d_1^B has been transferred to carrier A, who serves the two service demands of customer 1. Moreover, the two service demands of customer 2 have been interchanged, so carrier A serves d_2^B and carrier B serves d_2^A. The result is that carrier A performs one single route that serves a total demand of $10 = (d_1^A + d_1^B) + d_2^B$, and carrier B also performs one single route that serves a total demand of $10 = d_2^A + d_3^B$. The overall cost of the optimal solution with collaboration is $4 + 2\sqrt{2}$, with savings of about one third with respect to the solution without collaboration.

2.1. Maximum saving

We investigate here the maximum cost reduction that can be achieved thanks to the collaboration among carriers. We first give a lower bound for the optimal cost of the SCC-VRP and then show that such bound can indeed be achieved.

We denote by $z^*(\text{SCC-VRP})$ the optimal cost of the SCC-VRP, by $z^*(\text{VRP}_r)$ the minimum cost of the delivery routes of carrier $r \in C$, when it operates independently of all other carriers, that



(a) SCC-VRP Instance

(b) Optimal solution without collaboration

(c) Optimal solution with collaboration

Figure 1: SCC-VRP Example

is when carrier r serves its own customers N_r, and by $z^*(m\text{-VRP})= \sum_{r\in C} z^*(\text{VRP}_r)$ the overall cost when no collaboration exists. The optimal solution of each VRP_r can be obtained by solving a classical VRP over the set of customer N_r.

Obviously, the total cost with collaboration, $z^*(\text{SCC-VRP})$, cannot be smaller than the cost of each individual carrier and, in particular, the maximum cost of any carrier in the solution without collaboration. That is,

$$z^*(\text{SCC-VRP}) \geq \max_{r\in C} z^*(\text{VRP}_r).$$

Thus, $z^*(\text{SCC-VRP})$ cannot also be smaller than the average cost of a carrier without collaboration. Therefore,

Proposition 1.

$$z^*\,(\textit{SCC-VRP}) \geq \frac{z^*(m\textit{-VRP})}{m}.$$

The above relationship defines a lower bound for the optimal cost of the SCC-VRP that depends on the minimum total cost without collaboration. We will show now that it can indeed happen that the cost with collaboration $z^*(\text{SCC-VRP})$ equals $1/m$ times the total cost without collaboration $z^*(m\text{-VRP})$, reducing by a factor m the cost without collaboration.

Consider an instance with m carriers where all depots are co-located and all carriers share one customer and have no other customer. The distance between the depot and the customer is equal to 1. Let us assume that the demand of the customer for each carrier is very small and, in any case, such that the sum of the demands for all carriers does not exceed the capacity of the vehicle. Then, in the solution without collaboration each carrier has to visit the customer with a vehicle and $z^*(m\text{-VRP})= 2m$ whereas in the solution with collaboration only one vehicle will be sent to the customer and $z^*(\text{SCC-VRP})= 2$.

Clearly, this analysis shows the extreme case, that is the maximum possible saving. In the rest of the paper we will explore the average behaviour of a collaborative strategy.

3. MILP formulations for the SCC-VRP

In this section we propose two different MILP formulations for the SCC-VRP. The first one, that we call *vehicle-based* formulation, uses decision variables that describe explicitly the arcs traversed by the routes of each carrier. The second formulation, that we call *load-based* formulation, avoids the use of specific variables associated with each vehicle and is based on the load of the vehicles at the visited vertices. In each case, specific decision variables are needed to establish the specific demand orders of the customers served by each of the carriers.

We will use the following standard additional notation. For $S \subset V$, $\delta^+(S) = \{(i,j) \in A \mid i \in S,\ j \in V \setminus S\}$ denotes the set of arcs "leaving" S, and $\delta^-(S) = \{(i,j) \in A \mid i \in V \setminus S,\ j \in S\}$ denotes the set of arcs "entering" S. For a singleton we simply write $\delta(i)^+ = \delta^+(\{i\})$ and $\delta(i)^- = \delta^-(\{i\})$. For a subset $S \subset V$ and a carrier $r \in C$ the set of arcs of A^r in $\delta^+(S)$ and $\delta^-(S)$ are respectively denoted by $\delta_r^+(S) = \delta^+(S) \cap A^r$ and $\delta_r^-(S) = \delta^-(S) \cap A^r$. Finally, for a vector $y \in \mathbb{R}^{|A|}$ and a set of arcs $F \subset A$, we use the compact notation $y(F) = \sum_{a \in F} y_a$.

3.1. Vehicle-based formulation

For the *vehicle-based* formulation we extend the classical Vehicle Routing formulation with decision variables associated with arcs traversed by each of the vehicles [43].

For each carrier $r \in C$ we denote by K_r the index set of its (unlimited) fleet of homogeneous vehicles. We use two sets of decision variables: the routing variables x, which indicate the arcs traversed in each route, and the variables z, which indicate allocation of customer demands to carriers.

For each carrier $r \in C$, $k \in K_r$, $(i,j) \in A^r$, let x_{ij}^k be a binary routing variable, which takes the value 1 if arc (i,j) is used by vehicle k and 0 otherwise. As previously mentioned, these variables only need to be defined for arcs connecting pairs of customers sharing carrier r, or one such customer with depot o_r, as these are the only arcs that can be used in an optimal route associated with carrier r. For the allocation of customer demands to carriers, for $r \in C$, let z_{irs}^k be a binary variable that takes the value 1 if and only if the demand d_i^r of customer $i \in N_r$ is served by carrier $s \in C_i$ in route $k \in K_s$.

Then, the *vehicle-based* formulation (VF) for the SCC-VRP is:

$$
\begin{aligned}
(VF) \qquad \min \quad & \sum_{r \in C} \sum_{k \in K_r} \sum_{(i,j) \in A^r} c_{ij} x_{ij}^k && && (1)\\
\text{subject to} \quad & \sum_{s \in C_i} \sum_{k \in K_s} z_{irs}^k = 1 && i \in N,\ r \in C_i && (2)\\
& x^k(\delta_r^+(o_r)) \le 1 && r \in C,\ k \in K_r && (3)\\
& x^k(\delta_r^-(i)) - x^k(\delta_r^+(i)) = 0 && r \in C,\ k \in K_r,\ i \in N_r && (4)\\
& x^k(\delta_r^+(W)) \ge z_{isr}^k && r,s \in C, k \in K_r,\ W \subset N_r \setminus \{o_r\},\ i \in W \cap N_s && (5)\\
& \sum_{i \in N_r} \sum_{s \in C_i} d_i^s z_{isr}^k \le Q && r \in C,\ k \in C_r && (6)\\
& \sum_{i \in N_r} \sum_{s \in C_i} z_{isr}^{k-1} \ge \sum_{i \in N_r} \sum_{s \in C_i} z_{isr}^k && r \in C,\ k \in K_r \setminus \{\min_k K_r\} && (7)\\
& x_{ij}^k \in \{0,1\} && r \in C,\ (i,j) \in A^r,\ k \in K_r && (8)\\
& z_{irs}^k \in \{0,1\} && i \in N,\ r,s \in C_i,\ k \in K_s. && (9)
\end{aligned}
$$

The objective function (1) is the minimization of the total cost of all the arcs traversed by the routes of all carriers. Constraints (2) guarantee that each of the demands (for different carriers) of each customer are allocated to some of its carriers. Constraints (3)–(4) describe the flow balance on the nodes traversed by the vehicle routes. Constraints (5) play a double role. On the one hand, for singletons $W = \{i\}$ they guarantee that if the demand of customer i for carrier s is assigned to route $k \in K_r$ of carrier $r \in C_i$, then customer i is visited by some arc of that route. On the other hand, in general, they guarantee the connectivity of the routes with their depots by imposing that if some demand of a customer in a set W is visited by route k of carrier r, then that route must use at least one arc exiting from set W. Constraints (6) ensure that the capacity of the vehicles is not exceeded. Finally, constraints (7) partially avoid symmetry of the solutions by imposing that the k-th vehicle of each carrier is not used unless vehicle $k-1$ of carrier r is used as well. Moreover, we order the routes of each carrier by non-decreasing number of served customers. The domains of the variables are defined in (8)–(9).

Formulation (1)–(9) has $\sum_{r \in C} |A^r||K_r|$ variables x and $\sum_{i \in N, s \in C_i} |C_i|^2|K_s|$ variables z. Except for connectivity inequalities (5), the sizes of all other families of constraints are polynomial in the parameters of the problem. On the contrary, for a fixed $r \in C$, the size of family (5) is exponential in $|N_r|$.

Remark 1.

1. *Similarly to [24] constraints* (5) *can be replaced by the subtour elimination constraints*

$$\sum_{i,j \in W} x_{ij}^k \leq |W| - 1, \qquad \textit{for all } r \in C, k \in K_r, W \subseteq N_r \setminus \{o_r\}, \tag{10}$$

   which do not depend on the z variables and, together with (2)*, also guarantee that for each vehicle k of carrier r at least one arc leaves each vertex set W visited by k and not containing the depot o_r.*

2. *Any binary vector $\overline{z}$ satisfying constraints* (2), (6) *and* (7)*, can be extended to a solution $(\overline{x}, \overline{z})$, which is feasible for VF. For this, a series of Traveling Salesman Problems has to be solved, one for each activated route and allocated customers. These are still NP-hard problems.*

3. *Contrary to the previous item, not any binary vector $\overline{x}$ satisfying constraints* (3), (4)*, and* (10) *can be transformed into a feasible solution for VF, unless constraints imposing explicitly the assignment of customers demands to carriers are included.*

4. *Note that it is not enough to impose binary conditions on only one of the subsets of variables, x or z, so explicit binary conditions are needed both for the z and the x variables.*

3.2. Load-based formulation

As we will see in Section 5 VF is extremely demanding from a computational point of view. This can be explained by its high number of binary variables; note, in particular, the very high number of allocation variables z. This difficulty encouraged us to look for alternative formulations with a smaller number of binary variables. Here we introduce a formulation for the SCC-VRP that avoids the use of the vehicle index (k) for any design binary variable. This comes at the expenses of continuous load variables, controlling the total load of the routes on the traversed arcs, associated with a higher number of indices. *Load-based* formulations have been proposed in recent years by several authors [34, 7, 5] for different types of vehicle routing problems. In our case, in addition to the routing and load variables which are the usual ones in such models, we need specific assignment variables that keep track of the carriers that serve each of the customer demands.

The new routing variables x_{ij}^r take value 1 if arc $(i,j) \in A^r$ is used by a route of carrier $r \in C$, and 0 otherwise. Note that, since these variables are associated with carriers, for $r \in C$ fixed, $(x_{ij}^r)_{(i,j) \in A^r}$ represent all the routes of carrier r in an aggregated fashion. For the additional demand allocation variables we are also able to drop the index associated with vehicles. Now we use decision variables z_{irs}, with $i \in N$ and $r, s \in C_i$, to indicate whether or not the demand d_i^r is served by carrier s. The role of the additional set of load variables is to guarantee that vehicle capacities are not violated. For this it is crucial to guarantee a correct redistribution of incoming flows at the different nodes. The usual technique would be to associate load variables with the different routes. However, this is no longer valid in our case, unless we also associate with the routes the design variables x and z, which is precisely what we are trying to avoid. On the other hand, if the load variables are only associated with the carriers, we would not guarantee that incoming flows are correctly redistributed. To avoid this we use load variables associated with origin/destination pairs. In particular, we define continuous load variables l_{ij}^{rh}, with $r \in C$, $h \in N_r$, $(i, j) \in A^r$, which indicate the load served by carrier r to customer h that circulates through arc (i, j). With the above decision variables, the *load-based* formulation (LF) is the following:

$$
\begin{array}{llll}
(LF) \quad \min & \displaystyle\sum_{r\in C}\sum_{(i,j)\in A^r} c_{ij}x_{ij}^r & & (11)\\
\text{subject to} & \displaystyle\sum_{s\in C_i} z_{irs} = 1 & i \in N,\ r \in C_i & (12)\\
& x^r(\delta_r^+(i)) - x^r(\delta_r^-(i)) = 0 & i \in N,\ r \in C_i & (13)\\
& x^s(\delta_s^+(i)) \geq z_{irs} & i \in N,\ r, s \in C_i & (14)\\
& l^{rh}(\delta_r^+(o_r)) = \displaystyle\sum_{s\in C_h} d_h^s z_{hsr} & r \in C, h \in N_r & (15)\\
& l^{rh}(\delta_r^+(i)) - l^{rh}(\delta_r^-(i)) = \begin{cases} 0, & \text{if } h \neq i \\ -\sum_{s\in C_i} d_i^s z_{isr}, & \text{if } h = i \end{cases} & r \in C\ , i, h \in N_r & (16)\\
& \displaystyle\sum_{h\in N_r} l_{ij}^{rh} \leq Q x_{ij}^r & r \in C,\ (i,j) \in A^r & (17)\\
& x_{ij}^r \in \{0,1\} & r \in C,\ (i,j) \in A^r, & (18)\\
& z_{irs} \in \{0,1\} & i \in N,\ r, s \in C_i & (19)\\
& l_{ij}^{rs} \geq 0 & r \in C, s \in N_r, (i,j) \in A^r. & (20)
\end{array}
$$

Like in the VF, the objective (11) represents the total routing costs. Constraints (12) guarantee that all the demands are allocated to some carrier. Constraints (13) describe the flows for the aggregated routes for each carrier. In particular, these constraints ensure at each customer a balance on the number of incoming and outgoing routes from each carrier. Quite similarly to constraints (5), constraints (14) relate variables x and z and guarantee that if the assignment is made to a given carrier (z_{irs}), some routing variable associated with that carrier (x_{ij}^s) is activated. Constraints (15)–(17) are the flow balance constraints for the load variables, and update the loads through the arcs according to the served demands. In particular, constraints (15) impose that the overall load of all the routes starting at depot o_r for customer h coincides with the total demand served in the routes of carrier r for customer h. In turn constraints (16) impose that the load served by carrier r with destination at customer i delivered at customer h is zero, unless $h = i$. In such a case the delivered load is $\sum_{s\in C_i} d_i^s z_{isr}$. The relation between the routing and load variables is modeled by constraints (17), which also guarantee that the capacity of the vehicles is not exceeded.

Finally, the domains of the variables are given in (18)–(20).

Formulation (11)–(20) has $\sum_{r\in C} |A^r|$ binary routing variables x. The number of demand allocation variables z is $\sum_{i\in N} |C_i|^2$. The overall number of continuous load variables l, is $\sum_{r\in C} |N_r||A^r|$. As for the number of constraints, now the sizes of all families are polynomial in the parameters of the problem.

Therefore, in terms of number of variables and constraints LF should be preferred to VF, as it has a smaller number of both binary variables and constraints. On the other hand, it is well-known that the linear programming (LP) bound of *vehicle-based* formulations is tighter than that of *load-based* formulations [34]. In Section 5 we will compare empirically VF and LF and we will see that, in practice, the size of the formulation is more important.

4. Solution methodology

This section presents the methodological aspects of the proposed exact solution algorithms for VF and LF. A reason for solving the SCC-VRP exactly is that, being a new optimization problem, optimal solutions of as large as possible instances are required for assessing the quality of any heuristic. Even more importantly, by solving the SCC-VRP exactly it is possible to quantify precisely the savings that can be achieved through collaboration, by comparing the minimum cost of the SCC-VRP with the sum of the minimum costs faced independently by the carriers when no collaboration takes place. A heuristic solution of the SCC-VRP and an exact solution of the individual optimization problems of the carriers without collaboration would provide only a lower bound on the savings, while heuristic solutions of the optimization problems with and without collaboration would not allow us to assess the value of a collaborative solution.

The section starts by describing several families of valid inequalities for each formulation, and their corresponding separation algorithms. Then, a detailed description of the branch-and-cut algorithm for each formulation is provided.

4.1. Valid Inequalities for VF

Below we present some families of valid inequalities for VF.

- Cover inequalities
  We can derive valid cover-type inequalities (see, for instance, [37]) associated with the capacity constraints (6). A pair $(S, \{\overline{C}_i\}_{i\in S})$ with $S \subset N_r$ and $\overline{C}_i \subset C_i$ for all $i \in S$, defines a *cover* for carrier $r \in C$, if $\sum_{i\in S}\sum_{s\in\overline{C}_i} d_i^s > Q$. The *cover inequality* associated with a cover for carrier r is
  $$\sum_{i\in S}\sum_{s\in C_i} z_{isr}^k \leq \sum_{i\in S} |\overline{C}_i| - 1. \tag{21}$$
  As usual, the tightest cover inequalities (21) are associated with minimal covers, i.e. the ones that do not contain any cover.

- Capacity-cut inequalities
  Capacity-cut inequalities (CCIs) and their extensions have been widely used in vehicle routing problems (see, for instance, [28, 34, 36]). These constraints impose that any feasible solution must traverse a minimum number of arcs in the cut-set associated with a given set of customers

W; this number depends on the overall demand of the set W, $d(W)$, and on the vehicles capacity Q. However, typical CCIs are no longer valid for the SCC-VRP, neither considering individually each of the routes of a given carrier or aggregating all of them. The reason is one of the main characteristics of the SCC-VRP, which, in turn, becomes one of its main difficulties, namely that the amount of demand of each customer that will be actually served by each carrier is unknown in advance. Still, we can derive *ad-hoc* CCIs, closer in spirit to Generalized Large Multistar inequalities [28, 34], which also combine the rationale of the connectivity and capacity constraints (5)-(6) but remain valid for the SCC-VRP, in particular, for VF. Our CCIs impose, for each carrier, a minimum number of arc traversals in the cut-set associated with a given set of customers W, depending on the overall demand of the set W *served by the carrier* and the vehicles capacity Q. In particular, consider a carrier $r \in C$ and $W \subset N_r$. The overall demand of the customers of W served by carrier r is $\sum_{i \in W} \sum_{s \in C_i} d_i^s \left(\sum_{k \in K_r} z_{isr}^k\right)$. Thus, taking into account the vehicles capacity Q, the number of arcs of $\delta^+(W)$ traversed by the vehicles of carrier r must be at least $\left\lceil \frac{\sum_{i \in W} \sum_{s \in C_i} d_i^s \left(\sum_{k \in K_r} z_{isr}^k\right)}{Q} \right\rceil$. Hence, the following inequality is valid for VF:

$$\sum_{k \in K_r} x^k(\delta_r^+(W)) \geq \left\lceil \frac{\sum_{i \in W} \sum_{s \in C_i} d_i^s \left(\sum_{k \in K_r} z_{isr}^k\right)}{Q} \right\rceil. \tag{22}$$

Unfortunately, the right hand side of inequality (22) is non-linear, although the inequality can be substituted by the weaker linear inequality

$$\sum_{k \in K_r} x^k(\delta_r^+(W)) \geq \frac{\sum_{i \in W} \sum_{s \in C_i} d_i^s \left(\sum_{k \in K_r} z_{isr}^k\right)}{Q}. \tag{23}$$

- Symmetry breaking inequalities

  The inequalities below can be used to partially break the symmetry of the solutions produced by VF. Similar inequalities have been proposed in [8]

$$x_{j,o_r}^r \leq \sum_{i<j} x_{o_r,i}^r, j \in N_r, r \in C. \tag{24}$$

4.2. Valid Inequalities for LF

Below we present some families of valid inequalities for LF.

- Connectivity inequalities

  In LF the connectivity of the routes of the carriers with their corresponding depots is guaranteed by the flow balance constraints and the relation between the x and l variables. Therefore the following inequalities:

$$x^r(\delta^+(W)) \geq z_{isr} \tag{25}$$

are satisfied for all $r, s \in C$, $i \in C_s$, even if they are not explicitly stated in the formulation. Nevertheless, when the integrality of the x variables is relaxed, LF may produce fractional solutions that do not satisfy the above connectivity constraints (25), which are valid for LF and can be used to reinforce this formulation.

- Capacity-cut inequalities

  Capacity-cut inequalities can also be derived for LF. Recall that constraints (15) indicate that the overall load in the arcs leaving the depot of a given carrier $r \in C$ is precisely the total demand served by that carrier. Thus taking into account the vehicles capacity Q we can also deduce a lower bound for the number of arcs that must leave that depot, which must be at least $\left\lceil \frac{\sum_{i \in N_r} \sum_{s \in C_i} d_i^s z_{isr}}{Q} \right\rceil$. Hence, we obtain the following capacity-cut inequality, which is valid for LF:

$$x^r(\delta_r^+(o_r)) \geq \left\lceil \frac{\sum_{i \in N_r} \sum_{s \in C_i} d_i^s z_{isr}}{Q} \right\rceil. \tag{26}$$

  As in the analogous capacity-cut inequalities for VF, the right hand side of inequality (26) is non-linear, although we can substitute the inequality by the weaker linear inequality

$$x^r(\delta_r^+(o_r)) \geq \frac{\sum_{i \in N_r} \sum_{s \in C_i} d_i^s z_{isr}}{Q}. \tag{27}$$

  Note that the expression of the weaker inequality (27) associated with each carrier is quite simple, so it can be directly be incorporated to LF to reinforce it.

- Symmetry breaking inequalities

  Quite similarly to (24), inspired by [8], the inequalities below can be used to partially break the symmetry of the solutions produced by LF:

$$x_{jo_r}^r \leq \sum_{i \leq j} x_{o_r i}^r \quad j \in N_r, r \in C. \tag{28}$$

4.3. Separation algorithms

With the exception of the symmetry breaking inequalities, the number of inequalities in each of the families of valid inequalities introduced above is exponential on the number of customers. Hence, in order to use them within an algorithmic framework it is necessary to know how to solve the separation problem in each case. Below we address this issue for the proposed families of valid inequalities.

4.3.1. Separation of connectivity inequalities (5)

Let $(\overline{x}, \overline{z})$ denote the current LP solution and, for each carrier $r \in C$ and vehicle $k \in K_r$, $\overline{x}^k$ the partial LP solution associated with vehicle k, i.e. the components of $\overline{x}$ associated with k. Furthermore, $G_{\overline{x}}^k = (V^k, A_{\overline{x}}^k)$ denotes the support graph of the partial solution $\overline{x}^k$ for vehicle $k \in K_r$, $r \in C$, obtained from G by eliminating all arcs in A^r with $\overline{x}_{ij}^k = 0$ and all vertices that are not incident with any arc of $A_{\overline{x}}^k$.

Exact separation For each carrier $r \in C$ and vehicle $k \in K$, we identify min-cuts in $G_{\overline{x}}^k$ relative to the capacities $\overline{x}_a^k$ for all $a \in A_{\overline{x}^k}$. In particular, for each carrier $s \in C$ and customer $i \in N_s$ with $\overline{z}_{isr}^k > 0$ we find the minimum cut $\delta^+(W)$ separating i and o_r. If the value of the min-cut is smaller than $\overline{z}_{isr}^k$ then the inequality (5) associated with W, $r, s \in C$ $k \in K_r$, and customer $i \in N_s$ is violated by $(\overline{x}^k, \overline{z}^k)$. This separation is exact and similar to procedures that have been used by other authors to separate connectivity constraints for other node and arc routing problems [3, 4, 13].

Note that the exact separation described above can be quite time consuming as it requires to solve a max-flow problem for each $r, s \in C$, $k \in K_r$, $i \in N_s$ with $\overline{z}^k_{isr} > 0$. Even if the complexity of each max-flow problem is polynomial (see, for instance, [25]), in practice, it may be preferable to use a heuristic separation of these constraints.

<u>*Heuristic separation*</u> The heuristic for the separation of (5) associated with a carrier $r \in C$ and vehicle $k \in K$, looks for connected components in the subgraph of $G^k_{\overline{x}}$, that contains only those arcs with values $\overline{x}^k > \varepsilon$, where ε is a given parameter. Then, we compute the real value of the cut associated with each connected component W, that does not contain depot o_r. If $\overline{x}^k(\delta^+(W)) < \max\{\overline{z}^k_{isr} : s \in C, i \in N_s\}$, the connectivity inequality (5) associated with W is violated by $(\overline{x}, \overline{z})$.

The complexity of this heuristic separation is indeed much smaller than the exact separation as it only requires to compute the connected components of the subgraphs of $G^k_{\overline{x}}$ induced by the considered value of the parameter ε. This can be efficiently done with any algorithm based on Recursive Deep First Search, which has complexity linear on the number on $|E^k| + |V^k|$ [29].

Observe that when the partial LP solution $\overline{x}$ is integer the above separation becomes exact for $\varepsilon = 0$, independently of whether or not the components of $\overline{z}^k$ are integer. That is, when $\overline{x}^k$ is integer, an inequality (5) violated by $(\overline{x}^k, \overline{z}^k)$ will be found if it exists.

4.3.2. Separation of cover cut inequalities (21)

The performance of the branch-and-cut algorithm can be enhanced by separating and incorporating violated cover inequalities (21). To identify a cover cut violated by the current LP solution $(\overline{x}, \overline{z})$ we adapt to our case the usual separation for cover cuts (see, for instance, [37]) as follows. For each carrier $r \in C$ and vehicle $k \in K_r$, we define an auxiliary problem, which uses binary decision variables w_{is} for all $i \in N$, $s \in C_i$. The subproblem is

$$(Cov^k_r) \qquad w^* = \min \quad \sum_{i \in N} \sum_{s \in C_i} (1 - \overline{z}^k_{isr}) w_{is} \tag{29}$$

$$\text{subject to} \quad \sum_{i \in N} \sum_{s \in C_i} d^s_i w_{is} > Q \tag{30}$$

$$w_{is} \in \{0, 1\} \qquad\qquad r \in C, k \in K_r. \tag{31}$$

It is easy to see that there is a cover inequality (21) violated by $(\overline{x}, \overline{z})$ if an only if $w^* < 1$. In this case, the cover that induces a violated cut is associated with the index set of the variables at value 1 in an optimal solution to (Cov^k_r). That is, for all $i \in N$, $\overline{C}_i = \{s_i \mid w_{is} = 1 \text{ in the optimal solution}\}$, and $S = \cup_{i \in N} \overline{C}_i$. The detected violated cuts are lifted to obtain tightest inequalities with the procedure developed in [37].

4.3.3. Separation of capacity-cut inequalities

The exact separation of capacity-cut inequalities is intricate, even for their linearized weaker version. In fact, no polynomial time algorithm is known for similar inequalities for other types of problems and even the heuristic separation becomes quite involved [34]. Hence, we have not considered them in our branch and cut algorithm as we already consider the connectivity constraints and the cover cuts independently.

4.4. *Solution algorithms*

In this section we present the solution algorithms we propose for solving the *vehicle-based* formulation (VF) and the *load-based* formulation (LF) introduced in Section 3. Below we describe the main elements of the solution algorithm used in each case.

4.5. *Branch-and-cut for the VF*

The VF for the SCC-VRP (1)-(9) has been solved with an exact branch-and-cut algorithm. As usual, at each node of the enumeration tree, the algorithm solves a relaxed Linear Programming (LP) formulation and *ad hoc* separation procedures are applied to detect relaxed constraints and valid inequalities violated by the current LP solution. The current LP formulation is reinforced by incorporating to it the detected violated cuts, and the reinforced formulation is resolved. If no violated cuts are detected, the algorithm selects a variable to branch on and the current node is substituted by the two nodes associated with the corresponding subproblems. Initially the family of connectivity constraints (5), which is of exponential size, is relaxed and only a small subset of such constraints is kept. This subset contains all the constraints (5) associated with singletons, i.e. $W = \{i\}$ with $i \in N_r \cap N_s$, $r, s \in C$, for all $k \in K_r$. Recall that this subset of connectivity constraints guarantees that each customer is visited by a route associated with the carrier that serves its demand.

4.6. *Enumeration algorithm for the LF*

The LF for the SCC-VRP (11)-(20) has also been solved with a branch-and-cut algorithm. The family of inequalities that we have used to reinforce LF is small. On the one hand no cover inequalities can be derived. On the other hand, as mentioned, LF does not contain any family of constraints of exponential size and the family of valid cuts (25) will be satisfied by any integer solution. Still, for fractional LP solutions $(\overline{x}, \overline{z})$ they can be separated with the same procedures presented in Section 4.3.1. This can be combined with the reinforcement of the initial LF with the simple family of capacity-cut inequalities (27) associated with each carrier.

5. Computational experiments

In this section we describe the computational experiments we have run to analyze and compare the formulations proposed in the previous section. The formulations have been implemented in the Optimization Programming Language `OPL` and solved with a tailored branch-and-cut algorithm described in Section 4, and based on the commercial software `CPLEX` 12.1. All experiments have been run on a PC limited to 1 thread running at 2.6GHz and 60GB of RAM. In all cases the computing time is limited to two hours.

We have run two types of experiments. The first ones focus on the effectiveness of the alternative formulations we have proposed and, in particular, on the limits of the instance size that can be solved. The second series of experiments aims at getting insights on the empirical performance of the SCC-VRP in terms of the savings that can be obtained, the level of collaboration that can be achieved and the potential benefits of the collaboration, proving insights for the ultimate goal of this paper.

The section is organized as follows. First, the sets of benchmark instances used in the experiments are described in Section 5.1. Section 5.2 discusses the effect of the different separation strategies described in Section 4 and compares the numerical results obtained with the two proposed formulations for one set of the benchmark instances. Section 5.3 gives the results of the LF on all the sets of benchmark instances. Sections 5.4 and 5.5 analyze the characteristics of the solutions provided by the SCC-VRP. First, the potential savings due to collaboration and other indicators are discussed in Section 5.4. Then, in Section 5.5 we extend the SCC-VRP to consider a more general model that allows us to evaluate the impact that transfer costs among depots have on the savings that can be obtained.

5.1. Benchmark instances

We generated two sets of benchmark instances ($S1$, $S2$), where all instances have two carriers (A, B). The first set ($S1$) consists of 12 test instances inspired by the instances proposed by Cordeau [14] for the Multiple Depot Vehicle Routing Problem (MDVRP). The second set ($S2$) consists of 100 instances where customers are located in a square of 100 units of edge.

For the instances in $S1$ we selected 12 two-depot instances from [14] for the MDVRP. The problem proposed here is new but close to the MDVRP as explained in Section 2. For that reason, we inspired some of the instances to already existing MDVRP instances, with their distances and capacities, and assigned one depot to each carrier. To obtain instances of reasonable size for the SCC-VRP, we limited the number of customers to values between 18 and 30, and used the corresponding data from the original instances. Then, each customer was declared shared with probability 0.25, and we split in two halves the demand of each shared customer between the two carriers, assigning one more unit of demand to the first carrier in case it is an odd number. Finally, each non-shared customer was assigned to one carrier: the first half of the non-shared customers to the first carrier, and the second half to the second carrier. The characteristics of the instances are summarized in Table 1. Column Q gives the capacity of the vehicles. Column $d(N_A)$ gives the total demand for carrier A and in brackets the quantity of this demand corresponding to shared customers. Column $d(N_B)$ gives the same information relative to carrier B, and $d(N)$ the overall demand of all customers. Then N, N_A and N_B give the total number of customers, the number of customers with demand for carrier A and the number of customers with demand for carrier B, respectively. Finally, column *Shared* gives the number of shared customers.

$S1$ instance	Q	$d(N_A)$	$d(N_B)$	$d(N)$	$\|N\|$ ($\|N_A\|$, $\|N_B\|$)	*Shared*
1	100	152 (42)	176 (39)	328	23 (12, 17)	6
2	100	201 (55)	196 (53)	397	29 (16, 19)	6
3	100	163 (15)	110 (13)	273	20 (14, 9)	3
4	100	176 (37)	136 (34)	312	23 (16, 14)	7
5	100	179 (53)	173 (52)	352	24 (16, 16)	8
6	200	61 (25)	213 (22)	274	21 (8, 17)	4
7	200	93 (50)	180 (48)	273	20 (10, 16)	6
8	200	189 (57)	210 (54)	399	30 (17, 19)	6
9	500	585 (113)	325 (111)	910	20 (14, 9)	3
10	500	524 (56)	386 (54)	910	20 (14, 10)	4
11	60	56 (20)	112 (20)	168	18 (8, 14)	4
12	60	40 (20)	136 (20)	176	20 (8, 17)	5

Table 1: Data summary of the $S1$ instances

The set of benchmark instances $S2$ consists of two subsets: subset $S2_R$, which contains 50 instances where customers are randomly located in the above-mentioned 100×100 square; and subset $S2_C$, which contains 50 instances with clustered customers, where each instance has between

3 and 5 clusters. Depots are located at two different extremes of the square, i.e. one is located at position (0,0) and the other one at (100,100). In each subset we generated 10 instances for each of the values of $|N| \in \{10, 15, 20, 25, 30\}$, where again each customer has a probability of 0.25 of being a shared customer. Capacity takes the following values: 100, 200, 300, 400 or 500, and demands were generated accordingly. Demands were generated from integer uniform distributions with different parameters, to generate groups of instances with smaller and higher demands. See details in the Appendix, Tables 11 and 12. The meaning of the columns is the same as in Table 1, with an extra column *Demand* that contains the parameters of the uniform distribution that were used to generate the demands in the given instance. Both sets of instances are available at http://mrocariu.github.io/code/.

5.2. Preliminary results with benchmark set S1

Below we describe the results we obtained in some preliminary experiments that we run with VF and LF under different settings, in order to set the best strategies for their corresponding solution algorithms and to compare their efficiencies.

5.2.1. Numerical results for VF

We compared several strategies for the solution of VF, particularly for the separation of the cover cuts in combination with the cuts generated by `CPLEX`. In all cases the initial formulation includes the subset of constraints (5) associated with singletons, i.e. $W = \{i\}$ with $i \in N_r \cap N_s$, $r, s \in C$, for all $k \in K_r$. The remaining constraints (5) are handled as lazy constraints, i.e., they are only separated at the nodes where the LP relaxation is integer. As mentioned, in that case the heuristic separation of Section 4.3.1 is exact.

First, we show the effect of `CPLEX` cuts on the VF. The left part of Table 2 compares the results for different cut parameter values for `CPLEX`: cuts applied freely, cover cuts forbidden and all cuts forbidden. All the experiments ended reaching the time limit. Under the column *Obj* the value of the best feasible solution found is presented, except when no feasible solution was found, showed by "-". In general, better feasible solutions are obtained when no `CPLEX` cuts are generated. On average, the best objective values found when `CPLEX` applies cuts freely are +27.2 % worse, and when only cover cuts are forbidden +19.3% worse. Therefore, for the following experiments with the VF all `CPLEX` cuts are deactivated.

Second, we evaluated the effect of lifted cover cuts to decide how often the separation procedure should be applied: only at the root node or also at some of the nodes of the enumeration tree. Results are presented in the right part of Table 2, where the value of the best feasible solution found is shown. All the experiments ended either reaching the time limit or reaching memory requirements limits (indicated in the table with an M). When lifted cover cuts are applied every 500 nodes, memory problems arise and, in general, the solutions obtained are not better than the ones obtained without lifted cover cuts. However, applying lifted cover cuts at the root node usually improves the results obtained without applying lifted cover cuts, for all but one instance. We conclude that the best performance of the VF is achieved when `CPLEX` cuts are deactivated and when lifted cover cuts are separated only at the root node.

5.2.2. Numerical results for LF

We have also run some preliminary experiments with LF and the benchmark set $S1$ for comparing different strategies for its solution. The strategies that we compared are the following:

	VF + CPLEX Cuts			VF + Lifted Cover Cuts	
	Freely	Cover forbidden	All forbidden	Root node	Every 500
$S1$	*Obj*	*Obj*	*Obj*	*Obj*	*Obj*
1	650.82	421.37	416.97	368.15	397.46
2	-	858.48	578.22	555.56	M
3	279.45	-	-	345.29	M
4	824.36	478.46	466.65	552.38	610.55
5	580.44	-	430.87	425.65	M
6	280.19	289.83	297.47	326.16	330.74
7	174.75	269.1	162.22	162.22	162.22
8	582.55	870.64	424.22	424.22	M
9	837.01	644.74	716.98	716.98	732.87
10	1040.29	670.01	727.81	727.81	598.24
11	524.87	522.93	507.10	507.1	509.06
12	839.02	957.08	820.92	820.92	820.92
	+27.2 %	+19.3 %	-	+1.0%	+2.8%

Table 2: Solutions for the VF with different CPLEX cut parameters and different frequencies of the lifted cover cuts separation

ST1 LF with no added cuts and default parameters for CPLEX (block of columns under LF + CPLEX Default).

ST2 LF with no added cuts and default parameters for CPLEX except for the selection of the branching variables. Branching first on fractional assignment variables z_{irs} is enhanced by assigning them a higher priority (block of columns under LF + Branching prio on z).

ST3 LF reinforced initially with the simple family of capacity-cut inequalities (27) associated with each carrier and with default parameters for CPLEX (block of columns under LF + capacity cuts (27)).

ST4 LF enhanced with separation of connectivity cuts (4.3.1) for fractional solutions and default parameters for CPLEX (block of columns under LF + connectivity cuts (25)).

A summary of the obtained results is presented in Table 3. Each block consists of three columns. Columns under *Obj* give the value of the best solution at termination, columns under *%Gap* the percentage gap between the values of the best solution found and the lower bound at termination, and columns under *T(s)* the computing time. The time in seconds needed to optimally solve the instances or *TL* when the time limit was reached before proving optimality. Sometimes the executions terminated because of insufficient memory. Such cases are identified with an (M) after the computing time.

Strategies ST1, ST2, and ST3 find an optimal solution for six of the benchmark instances, but fail to prove optimality of the best solution found within the maximum computing time for the remaining six instances. In all three strategies the instances solved to optimality are the same. Strategy ST4 proves the optimality of the best solution found for four of the instances. For the remaining eight instances it terminates without an optimal solution, because of insufficient memory with three instances, and because of the time limit with the remaining five instances. No substantial differences can be appreciated among the first three strategies: ST2 and ST3 seem to be a little faster for the instances solved to optimality, although the gaps for the unsolved instances seem to be a little better for ST1.

	LF + CPLEX Default			LF + Branching prio on z			LF + capacity cuts (27)			LF + connectivity cuts (25)		
	Obj	%Gap	T(s)	Obj	%Gap	T(s)	Obj	%Gap	T(s)	Obj	%Gap	T(s)
1	273.88	5.46	*TL*	274.05	8.58	*TL*	275.08	7.22	*TL*	273.77	8.61	*TL*
2	324.19	7.28	*TL*	323.76	6.51	*TL*	322.01	6.56	*TL*	498.01	41.79	7133 (M)
3	233.02	0.98	*TL*	233.63	5.19	*TL*	233.02	6.65	*TL*	292.10	30.62	4706 (M)
4	322.32	3.70	*TL*	322.30	9.90	*TL*	322.51	8.81	*TL*	322.51	12.72	*TL*
5	328.02	7.01	*TL*	326.94	7.12	*TL*	322.79	3.91	*TL*	323.77	12.04	*TL*
6	230.08	0.00	134	230.08	0.00	126	230.08	0.00	92	230.08	0.00	278.8
7	156.93	0.00	120	156.93	0.00	103	156.93	0.00	481	156.93	0.00	1279
8	237.83	0.00	2472	237.83	0.00	1718	237.83	0.00	690	359.91	37.29	*TL*
9	392.06	0.00	60	392.06	0.00	46	392.06	0.00	54	392.06	0.00	89
10	455.71	0.00	90	455.71	0.00	110	455.71	0.00	116	455.71	4.77	*TL*
11	486.90	0.00	727	486.90	0.00	726	486.90	0.00	712	486.90	0.00	1341
12	750.68	13.00	*TL*	749.84	12.26	*TL*	750.52	8.40	*TL*	751.08	14.57	*TL*

Table 3: Solutions for different strategies with the LF

The results obtained with all tested strategies indicate that effectiveness of the LF does not seem to be affected by the inclusion of valid inequalities or tailored cuts. This behavior is analogous to that of other *load-based* formulations for similar problems studied in recent papers (see [6]).

5.2.3. VF versus LF

Next we compare the results obtained with VF against the ones obtained with LF for the set of instances $S1$. For this analysis we compare the results produced by VF when CPLEX cuts are deactivated and lifted cover cuts are separated only at the root node, and those produced by LF with strategy ST1. The comparison is summarized in Table 4. Columns under *Obj* give the values of the best feasible solution found in each case. Columns r_A, r_B indicate the number of vehicles needed by each carrier in the optimal/best-known solution. Columns under $T(s)$ give the computing times (in seconds). These are the times needed to optimally solve the instances or *TL* when the time limit was reached before proving optimality.

	VF					LF				
	Obj	r_A	r_B	*%Gap*	$T(s)$	*Obj*	r_A	r_B	*%Gap*	$T(s)$
1	337.45	2	2	53.44	*TL*	273.88	2	2	5.46	*TL*
2	518.13	2	2	66.05	*TL*	324.19	2	2	7.28	*TL*
3	316.78	2	2	55.11	*TL*	233.28	2	1	0.98	*TL*
4	563.58	2	2	68.00	*TL*	322.3	2	2	3.7	*TL*
5	468.54	2	2	60.60	*TL*	328.02	2	2	7.01	*TL*
6	259.87	1	2	32.68	*TL*	230.08	1	2	0	134.11
7	180.56	1	1	35.81	*TL*	156.93	1	1	0	120.1
8	536.03	2	2	65.82	*TL*	237.83	1	1	0	2472.56
9	515.48	2	2	54.12	*TL*	392.06	1	1	0	59.9
10	685.95	2	1	65.05	*TL*	455.71	1	1	0	90.19
11	494.6	1	2	20.53	*TL*	486.9	1	2	0	726.98
12	882.65	1	2	56.54	*TL*	750.6	1	2	13.00	*TL*

Table 4: Solutions for the VF and the LF for instances $S1$.

As it can be seen, the results of the LF clearly outperform those of the VF. The LF is able to provide optimal solutions for 6 of the $S1$ instances, with maximum percentage optimality gaps of 13%. On the contrary, despite the efforts to reinforce the formulation and to separate violated cuts, none of the 12 tested instances could be optimally solved with the VF within the 2 hours time limit. Furthermore, the percentage optimality gaps and the percentage deviations of the VF solution with respect to the LF solution are quite large. Note that the smallest percentage gap at termination of the solutions produced by VF is 20%. We observe that, quite consistently, solutions proven to be optimal or with small percentage gaps involve a small number of routes for the carriers.

The obtained numerical results confirm empirically that the smaller number of binary variables of the LF determines its effectiveness in comparison to the VF. Therefore, all the experiments that we report in the following were run with the LF with strategy ST1.

5.3. LF with $S2$ instance set

Here we analyze the results of the LF for the 100 instances of set $S2$, which are summarized in Table 5. The table contains two blocks of columns, one for the *random* instances of $S2_R$ and another one for the *clustered* instances of $S2_C$. Within each block, each row gives aggregate or average results for the 10 instances of the same size in the block. The sizes of the instances are given in the first column (labeled $|N|$). Entries in columns *#Opt* give the number of instances in the group optimally solved with the formulation. Entries in columns *Gap(%)* give the average percentage optimality gaps. The last column in each block (*T(s)*) shows the average computing times in seconds.

	$S2_R$ *Random*			$S2_C$ *Clustered*		
$\|N\|$	*#Opt*	*%Gap*	*T(s)*	*#Opt*	*%Gap*	*T(s)*
10	10	0.00	12.11	10	0.00	5.67
15	10	0.00	412.63	9	0.23	807.70
20	5	2.95	3682.59	6	3.24	3546.46
25	4	5.94	4388.68	3	9.79	5112.67
30	1	10.69	6646.46	0	12.51	7200.00

Table 5: Summary of solutions for set $S2$

As it can be seen, all instances but one with up to 15 customers were optimally solved, for both the $S2_R$ and the $S2_C$ classes. A provable optimal solution was also found for the 5 $S2_R$ instances with 20 customers and 6 out of the 10 $S2_C$ instances of the same size. As expected, the number of optimally solved instances decreases as their size increases. In particular, only one instance with 30 customers for the class $S2_R$ is optimally solved. Nevertheless, for the instances that were not optimally solved, the percentage optimality gaps are relatively small and, on average 10.69% for the 30 customers instances of $S2_R$ and 12.51% for the $S2_C$ instances of the same size. In general, the *random* instances seem to be somehow less hard to solve than the *clustered* instances, as the average computing times and percentage optimality gaps are smaller. In order to obtain better managerial insights of the potential savings produced by the proposed model, in the following sections we further discuss this and other related issues.

5.4. Savings due to the collaboration

In this section we compare the solutions produced by the LF with the solutions of the same instances when no collaboration exists and each carrier serves all its customers independently from the other carriers. The optimal solution for each individual carrier is obtained by finding the optimal routes that visit all its customers. The cost of the solutions in the setting without collaboration is obtained summing up the individual costs of all the carriers. All VRPs without collaboration were optimally solved both for instances in S1 and S2. Even if for each collaborative instance, we need to solve two *individual* instances, one for each carrier, individual problems are easier to solve as they require less decisions, and also their size decreases considerably.

The two sets of instances, $S1$ and $S2$, are analyzed separately. Table 6 presents, for the 12 instances of $S1$, the results for the independent carriers (under Without collaboration (A,B)) and the results for the SCC-VRP (under Collaboration). Obj_A, Obj_B and Obj are the costs of the routes performed by carriers A and B in the solution without collaboration and of the collaborative solution, respectively. r_A and r_B give the number of routes and "- %" the savings. On some instances, the LF does not obtain optimal solutions, as previously shown in Table 4. Still, the obtained results allow us to appreciate the savings that could be obtained on these instances via collaboration, given that the savings obtained with suboptimal solutions are lower bounds of the actual savings that can be obtained with optimal solutions. Thus, the value of the best-known solution at termination has been used for the comparison, also in the cases where this solution was not proven to be optimal. The results of Table 6 show that the obtained cost savings range from a minimum of 6.5% to a maximum of 25.2%. The average cost reduction is 13.9%. Furthermore, in terms of number of vehicles, the collaborative solution allows us to reduce the fleet size in several instances.

$S1$	Without collaboration (A,B)					Collaboration			Savings		
Instance	Obj_A	Obj_B	$Obj_A + Obj_B$	r_A	r_B	Obj	r_A	r_B	- %	-$\%_A$	-$\%_B$
1	171.38	168.02	339.4	2	2	273.88	2	2	23.9	35.5	2.9
2	173.37	179.8	353.17	3	2	324.19	2	2	8.9	17.9	2.7
3	158.56	113.45	272.01	2	2	233.02	2	1	16.7	1.8	31.6
4	194.97	173.14	368.11	2	2	322.3	2	2	14.2	14.5	10.8
5	171.94	191.66	363.6	2	2	328.02	2	2	10.8	10.3	13.3
6	107.48	145.87	253.35	1	2	230.08	1	2	10.1	17.6	2.9
7	65.8	104.7	170.5	1	1	156.93	1	1	8.6	0	12.9
8	118.65	161.41	280.06	1	2	237.83	1	1	17.8	1.6	25
9	296.37	194.45	490.82	2	1	392.06	1	1	25.2	33.3	0
10	293.45	230.38	523.83	2	1	455.71	1	1	14.9	18.1	6.5
11	203.49	329.21	532.7	1	2	486.9	1	2	9.4	22.4	0
12	243.27	556.01	799.28	1	3	750.68	1	2	6.5	0	8.9

Table 6: Solutions for set $S1$ without and with collaboration

The set of instances $S2$ was similarly solved under the same settings. Table 7 summarizes the percentage savings obtained in each group of instances. Note that this is again a lower bound on the savings, even if some collaborative solutions were not proven to be optimal. The number of optimal solutions obtained in each set of instances is under column *#Opt*. Column "-%" gives the average cost savings in the set of instances.

Average cost savings range from 9.8 to 18.3 % in the random set $S2_R$ and from 2.5 to 17.8 % in the clustered set $S2_C$. Lower savings can be observed in the clustered instances of smaller sizes. This is due to the fact that collaboration does not bring significant benefits in clustered instances with few shared customers. When clusters have exclusive customers of both carriers, transferring shared customers will not avoid that both carriers visit those clusters. Higher savings are expected in instances of larger size (25 and 30). Unfortunately, we can only provide a lower bound on savings for these sizes since optimal solutions are unknown. In any case, except for the small clustered instances, savings are larger than 9.8 %.

Going further in the analysis, one question that arises is what costs and revenues are for each of the carriers in the collaborative solutions. Our results provide insights on this issue, even if a sophisticated analysis is beyond the scope of this paper where we highlight the potential benefits of the simplest form of collaboration among carriers. As mentioned, on average, in the collaborative solution there is a reduction of the overall routing costs. Hence, it is possible to split the overall savings in such a way that both carriers individually experience cost reductions. In fact, columns "-$\%_A$" and "-$\%_B$" of Tables 6 and 7 indicate that in each group of instances both carriers already experience savings, on average. However, in some cases individual savings are not balanced and do not reflect the value of transferred demand. Moreover, in some particular instances it is possible that a given carrier experiences an increase of its routing costs. In the following subsection 5.4.1 we discuss this situation for a particular instance.

	$S2_R$ Random				$S2_C$ Clustered			
N	$\#Opt$	-%	-% A	-% B	$\#Opt$	-%	-$\%_A$	-$\%_B$
10	10	13.4	7.9	13.7	10	2.5	2.5	2.5
15	10	12.0	12.6	6.4	9	7.3	1.2	11.1
20	5	18.3	19.2	11	6	17.8	8.3	18.6
25	4	9.8	8.4	10.9	3	11.1	6.1	12.6
30	1	15	20.5	8.1	0	11.4	14	7.5

Table 7: Average cost savings in the set of instances $S2$ with collaboration. Individual savings need to be computed

5.4.1. Cost compensation

In this section we discuss the results for one particular instance to get better insights on the current results and possible compensation mechanisms.

For the analysis we selected instance 1001 of the set $S2_R$ (see Table 11). Figure 2 shows the instance, which has four shared customers depicted in green, two exclusive customers of carrier A in red and five exclusive customers of carrier B in blue. The obtained solutions with collaboration are plotted on the left and the individual solutions without collaboration on the right. In the individual solution both carriers visit all the shared customers. Red carrier (A) needs one route to visit all customers, and blue carrier (B) uses two different routes. Instead, in the collaborative solution three out of the four shared customers are transferred from B to A. This makes carrier A build a new route because, due to capacity reasons, it cannot incorporate the additional demand in one single route. Instead, carrier B can now serve all its assigned demand with only 1 route. In terms of costs, the overall cost is reduced by 25.65 %. However, looking at individual costs, carrier A suffers an increase of 22.9 % and only carrier B has a decrease of 44.52 %. These costs reflect the transfer of shared customers from carrier B to carrier A. Table 8 summarizes the features of the solution. Columns Dem_{AB} and Dem_{BA} give the fraction of the overall demand corresponding to shared customers transferred from A to B and from B to A, respectively. Unlike what happens in most cases, in this particular instance, carrier A experiences a considerable increase of cost.

	Obj_A	Obj_B	Obj	r_A	r_B	r_T	Dem_{AB}	Dem_{BA}
Without collaboration	249.71	448.77	698.48	1	2	3	-	-
Collaboration	306.89	248.99	555.89	2	1	3	0	0.8
Variation	+22.90	-44.52	- 25.65	+1	-1	=	-	-

Table 8: Solution data for instance 1001 with and without collaboration

With the above information, a compensating mechanism should be adopted, based on the shared

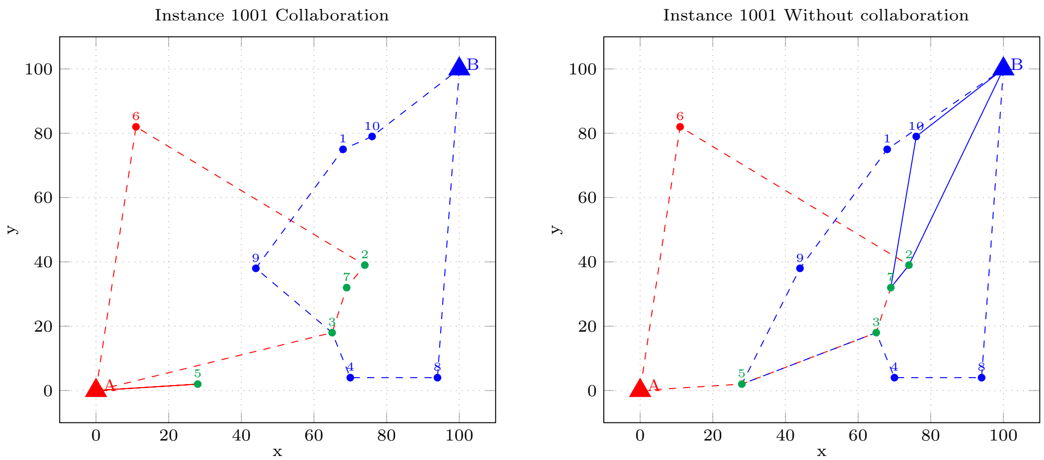


Figure 2: SCC-VRP Solution 1001 with and without collaboration

customers transferred and the final costs of the collaborative solution, that can be attractive for both carriers. For instance, constraints could be added to the formulation to guarantee that no carrier performs routes with greater costs than in the case without collaboration. Such an alternative was proposed in [22] for the collaboration uncapacitated arc routing problem. Other possibilities could take into account the proportion of shared demand of each carrier transferred to the other carrier. In its simplest form, the overall saving, Sav, would be split among carriers A and B in quantities Sav_A and Sav_B, respectively, such that $Sav = Sav_A + Sav_B$, and $Sav_A/Dem_{BA} = Sav_B/Dem_{AB}$, where Dem_{AB} (resp Dem_{BA}) denote the fraction of shared demand of carrier A (resp. B) which has been transferred to carrier B (resp. A). Note that in the case that one carrier does not serve any transferred demand from the other (i.e. Dem_{AB} or Dem_{BA} are zero), the previous formula is not valid. For this case we suggest that the carrier that does not serve any transferred demand gets a smaller portion of the savings compared to the one serving the transferred demand. For instance, if $Dem_{AB} = 0$ and $Dem_{BA} \neq 0$, then $Sav_B = kSav_A$ with $k \in [0, 1)$. This guarantees that the company serving the transferred demand (A) gets a bigger portion of the savings, but the other company (B) receives a part of the savings too.

5.4.2. Sharing percentage

In this section we describe the outcome of a specific experiment we performed in order to study the impact that the percentage of shared customers has on the potential savings. For the experiment, one test instance with 15 shared customers was specifically created. Then, the same instance was solved under different sharing assumptions. First, the instance was solved without collaboration. Then, the instance was solved under all intermediate levels of collaboration, i.e., assuming one, two, three, ... customers are shared. Finally, the instance was also solved under total collaboration circumstances, i.e. carriers share all customers.

In the above experiment, some of the instances could not be optimally solved within the allowed computing time of two hours. This happened for the cases with a smaller percentage of shared customers. Note that the size of the instances increases as the number of customers that cannot be shared increases, as such customers must be represented as two co-located non-shared customers, each of them with demand for one carrier.

Figure 3 gives the objective function values for the different percentages of shared customers. In the cases where optimal solutions could not be obtained, best-known results were used. The obtained results clearly indicate that, as expected, higher savings can be obtained as the percentage of shared customers increases.

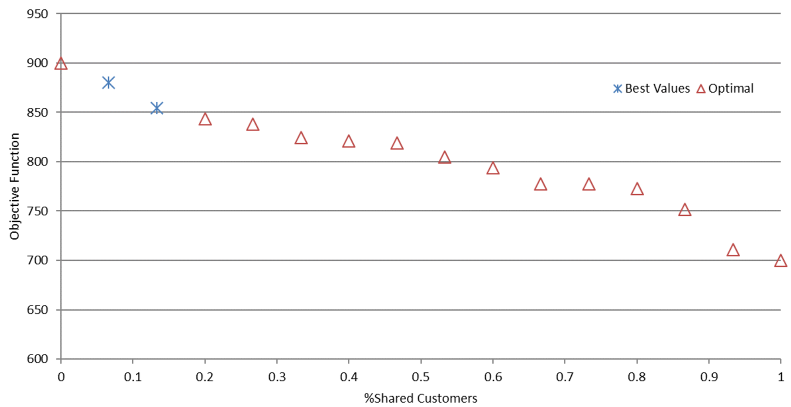


Figure 3: Objective function values for different percentage of shared customers. Instance with 15 clustered customers

5.5. The cost of demand transfer

In the previous section we focused on the potential savings achieved from carriers collaboration. Nevertheless, it is clear that collaboration may also imply some logistics costs. For instance, when the products distributed by the different carriers are not the same, round trips connecting the depots of the carriers may be needed to make the necessary amount of transferred demand available to the serving carrier. Note, however, that these costs do not always apply. For instance, when the depots of the carriers are co-located, as it is the case when carriers operate from the same consolidation center.

Below we extend the LF to include the transfer cost among carriers when they arise. We assume that at most one round-trip is needed to connect each pair of depots (i.e., sufficiently large vehicles are used). Thus, we consider a fixed set-up cost (F) for each round-trip between a pair of depots. Then, we define the following set of binary variables to determine whether or not a trip takes place between a given pair of depots. Let y_{rs}, $r, s \in C, s > r$, be a binary variable that takes value 1 if and only if some demand of carrier r is served by carrier s or vice versa. Let v_C denote the overall number of round-trips between depots. Then, in order to account for the transfer costs, the objective function is extended to:

$$\min \sum_{r \in C} \sum_{(i,j) \in A} c_{ij} x_{ij}^r + F v_C. \tag{32}$$

In addition, the following sets of constraints must be added to relate the demand allocation variables to the new transfer variables:

$$z_{irs} \leq y_{rs} \qquad i \in N,\ r, s \in C_i, s > r \tag{33}$$

$$z_{isr} \leq y_{rs} \qquad i \in N,\ r, s \in C_i, s > r \tag{34}$$

$$v_C = \sum_{r \in C} \sum_{s \in C, s > r} y_{rs} \tag{35}$$

$$y_{rs} \text{ binary}, \qquad r,\ s \in C\ ,\ s > r \tag{36}$$

$$v_C \text{ integer}. \tag{37}$$

Constraints (33)–(34) relate the z and y variables, by activating round-trip connections between pairs of depots when there is a demand transfer between the corresponding carriers. Constraint (35) simply counts the overall number of round-trips. The domains of the variables, and binary/integer conditions, are given in (36)–(37).

We present now the results obtained to analyze the tradeoff between transfer costs and potential savings. In Tables 9 and 10 we compare the results of three experiments, one for each value to $F \in \{0, 20, 50\}$. We solve the collaborative problem that includes transfer costs and the problem without collaboration. Column *#Opt* shows the number of instances solved to optimality in the collaborative case. Column *#Col* shows the number of instances where the optimal (or best-known) collaborative solution allows a cost reduction with respect to the solution without collaboration. The saving is shown under "-%". As expected, the results in Tables 9 and 10 indicate that almost in all cases the solution is exactly the same when the transfer cost is small ($F = 20$). In such cases the only difference is in the value of the objective function, which increases by 20 units (the cost of the round-trip between the two depots). For $F = 20$ there are, however, some cases where the savings due to collaboration do not compensate the increase of 20 units due to the round-trip and the solution without collaboration is best. Of course, this behavior becomes more frequent as the transfer cost increases, as can be seen in the block of columns with $F = 50$. Both tables also show the average computing times in seconds. Note that these are comparable with the ones obtained without transfer cost presented in Table 5.

	Transfer cost = 0			Transfer cost = 20				Transfer cost = 50			
N	*#Opt*	*#Col*	-%	*# Opt*	*#Col*	*T(s)*	-%	*#Opt*	*#Col*	*T(s)*	-%
10	10	10	13.4	10	10	20.99	9.1	10	7	14.19	4.6
15	10	10	12.0	10	9	720.92	8.6	10	7	414.52	4.4
20	5	10	18.3	6	10	3540.08	14.8	5	10	3723.96	10.0
25	4	10	9.8	4	10	4385.97	7.4	4	7	4405.88	4.1
30	1	10	15	2	9	6375.33	12.8	2	9	6485.68	7.9

Table 9: Comparison of solutions with and without collaboration with transfer cost 0, 20 and 50 in $S2_R$.Colaboration index should be revised

	Transfer cost = 0			Transfer cost = 20				Transfer cost = 50			
N	*#Opt*	*#Col*	-%	*# Opt*	*#Col*	*T(s)*	-%	*#Opt*	*#Col*	*T(s)*	-%
10	10	10	2.5	10	1	6.13	0.4	10	0	4.34	0.0
15	9	10	7.3	9	5	954.75	4.6	9	4	938.62	2.5
20	6	10	17.8	5	9	3729.4	13.8	6	7	3668.65	9
25	3	10	11.1	3	10	5081.65	7.8	3	4	5097.87	3.4
30	0	10	11.4	0	10	7200	8.3	1	8	7018.83	5.7

Table 10: Comparison of solutions with and without collaboration with transfer cost 0, 20 and 50 in $S2_C$. Colaboration index should be revised

6. Conclusions

In this paper the Shared Customer Collaboration Vehicle Routing Problem (SCC-VRP), a new model for horizontal collaboration in the framework of last-mile deliveries, was introduced. The main goal of the paper is to assess the benefits of a collaborative approach to freight distribution in the context of urban environments.

From the methodological point of view, it was shown that for the new collaborative model a *load-based* formulation is more effective than a *vehicle-based* formulation. The minimum cost that can be achieved through collaboration, obtained through the optimal solution of the SCC-VRP, was compared to the sum of the costs of the carriers in case they work independently from each other, that is without collaboration. The solution without collaboration was obtained by solving a classical Vehicle Routing Problem for each of the carriers. Whereas the saving factor due to collaboration may be as large as the number of carriers, the saving computed on a set of benchmark instances solved to optimality depends on the number of shared customers and on their location and ranges from 6.5 % to 25.2 %.

The proposed model can be used to assess the potential benefits of collaboration among independent carriers that may, thanks to the computed benefits, be motivated to form a coalition and put in place a collaborative scheme. It may also be used to optimize the operations of an already formed coalition. A collaborative scheme among carriers may be of crucial importance in a competitive environment, especially when carriers are of small or medium size. With this paper we intend to stimulate research on collaborative schemes that, besides making carriers more competitive, can contribute to reduce the freight distribution costs thanks to an overall reduced distance travelled, a higher average load, and a smaller number of used vehicles. Such reduction of operational costs in turn implies a lower level of environmental impact of freight distribution.

Several research directions remain to be explored. Heuristics should be designed for the proposed SCC-VRP. Moreover, extensions of the proposed model include accurate modeling of the costs of transferring goods between depots, location decisions for the depots or for a joint consolidation center, a multi-period setting. A particularly important and challenging research direction concerns the design and analysis of compensation schemes that should guarantee that each carrier has benefits from the collaboration and be fair to all carriers.

Acknowledgements

The research of the first two authors has been partially supported by the Spanish Ministry of Economy and Competitiveness through grant MTM2015-63779-R, and ERDF funds. The research of the first author is partially supported by MINECO/FEDER grant MDM-2014-044. This support is gratefully acknowledged. The authors also acknowledge three reviewers whose comments and suggestions helped them improve a previous version of this paper.

7. Appendix

7.1. Instance set $S2_R$

Instance	R/C	Q	Demand	$d(N_A)$	$d(N_B)$	$d(N)$	$\|N\|$ ($\|N_A\|$, $\|N_B\|$)	*Shared*
1001	R	100	$U\sim[5,20]$	76(59)	147(65)	223	10 (5, 9)	4
1002	R	200	$U\sim[5,20]$	42(19)	102(24)	144	10 (4, 8)	2
1003	R	300	$U\sim[25,40]$	90(45)	127(47)	217	10 (6, 7)	3
1004	R	400	$U\sim[25,40]$	104(51)	91(43)	195	10 (7, 6)	3
1005	R	500	$U\sim[25,50]$	201(61)	178(61)	379	10 (6, 6)	2
1006	R	100	$U\sim[5,20]$	154(53)	72(53)	226	15 (13, 6)	4
1007	R	200	$U\sim[5,20]$	123(62)	108(30)	231	15 (9, 10)	4
1008	R	300	$U\sim[25,40]$	149(64)	159(50)	308	15 (8, 10)	3
1009	R	400	$U\sim[25,40]$	144(30)	148(37)	292	15 (9, 8)	2
1010	R	500	$U\sim[25,50]$	162(35)	332(31)	494	15 (5, 11)	1
1011	R	100	$U\sim[5,20]$	139(46)	153(48)	292	20 (12, 12)	4
1012	R	200	$U\sim[5,20]$	173(73)	150(58)	323	20 (13, 13)	6
1013	R	300	$U\sim[25,40]$	188(98)	234(102)	422	20 (12, 14)	6
1014	R	400	$U\sim[25,40]$	192(111)	277(126)	469	20 (11, 16)	7
1015	R	500	$U\sim[25,50]$	451(238)	415(238)	866	20 (15, 13)	8
1016	R	100	$U\sim[5,20]$	208(80)	202(79)	410	25 (16, 15)	6
1017	R	200	$U\sim[5,20]$	223(31)	160(47)	383	25 (16, 12)	3
1018	R	300	$U\sim[25,40]$	253(70)	247(77)	500	25 (14, 15)	4
1019	R	400	$U\sim[25,40]$	263(82)	301(75)	564	25 (13, 16)	4
1020	R	500	$U\sim[25,50]$	551(42)	339(22)	890	25 (16, 10)	1
1021	R	100	$U\sim[5,20]$	285(120)	176(100)	461	30 (24, 15)	9
1022	R	200	$U\sim[5,20]$	287(134)	261(168)	548	30 (22, 19)	11
1023	R	300	$U\sim[25,40]$	383(65)	225(70)	608	30 (21, 13)	4
1024	R	400	$U\sim[25,40]$	375(177)	369(161)	744	30 (20, 19)	9
1025	R	500	$U\sim[25,50]$	674(281)	573(273)	1247	30 (19, 19)	8
1026	R	100	$U\sim[5,35]$	151(91)	121(57)	272	10 (7, 7)	4
1027	R	200	$U\sim[5,35]$	66(37)	138(44)	204	10 (4, 8)	2
1028	R	300	$U\sim[25,50]$	222(95)	296(128)	518	10 (6, 7)	3
1029	R	400	$U\sim[25,50]$	230(116)	238(85)	468	10 (6, 7)	3
1030	R	500	$U\sim[25,75]$	197(156)	401(191)	598	10 (5, 9)	4
1031	R	100	$U\sim[5,35]$	149(14)	117(17)	266	15 (9, 7)	1
1032	R	200	$U\sim[5,35]$	103(40)	211(36)	314	15 (6, 11)	2
1033	R	300	$U\sim[25,50]$	349(195)	404(179)	753	15 (9, 11)	5
1034	R	400	$U\sim[25,50]$	390(59)	247(63)	637	15 (10, 7)	2
1035	R	500	$U\sim[25,75]$	601(189)	398(136)	999	15 (10, 8)	3
1036	R	100	$U\sim[5,35]$	235(61)	243(66)	478	20 (12, 11)	3
1037	R	200	$U\sim[5,35]$	268(96)	217(81)	485	20 (14, 11)	5
1038	R	300	$U\sim[25,50]$	510(201)	520(239)	1030	20 (13, 12)	5
1039	R	400	$U\sim[25,50]$	488(173)	448(160)	936	20 (12, 12)	4
1040	R	500	$U\sim[25,75]$	611(187)	567(247)	1178	20 (13, 11)	4
1041	R	100	$U\sim[5,35]$	343(64)	169(37)	512	25 (16, 11)	2
1042	R	200	$U\sim[5,35]$	298(88)	340(127)	638	25 (16, 14)	5
1043	R	300	$U\sim[25,50]$	362(81)	704(70)	1066	25 (9, 18)	2
1044	R	400	$U\sim[25,50]$	662(190)	482(152)	1144	25 (16, 14)	5
1045	R	500	$U\sim[25,75]$	773(298)	745(272)	1518	25 (16, 15)	6
1046	R	100	$U\sim[5,35]$	418(238)	463(222)	881	30 (20, 20)	10
1047	R	200	$U\sim[5,35]$	313(108)	352(105)	665	30 (18, 17)	5
1048	R	300	$U\sim[25,50]$	691(360)	789(355)	1480	30 (18, 22)	10
1049	R	400	$U\sim[25,50]$	565(343)	855(374)	1420	30 (16, 23)	9
1050	R	500	$U\sim[25,75]$	1095(432)	892(455)	1987	30 (22, 17)	9

Table 11: Data summary of set $S2_R$

7.2. Instance set $S2_C$

Instance	R/C	Q	Demand	$d(N_A)$	$d(N_B)$	$d(N)$	$\|N\|$ $(\|N_A\|, \|N_B\|)$	*Shared*
1051	C	100	$U \sim [5, 20]$	97(44)	85(38)	182	10 (7, 6)	3
1052	C	200	$U \sim [5, 20]$	45(27)	126(45)	171	10 (5, 8)	3
1053	C	300	$U \sim [25, 40]$	79(52)	127(37)	206	10 (5, 8)	3
1054	C	400	$U \sim [25, 40]$	110(22)	72(16)	182	10 (6, 5)	1
1055	C	500	$U \sim [25, 50]$	174(50)	181(56)	355	10 (6, 6)	2
1056	C	100	$U \sim [5, 20]$	124(40)	134(47)	258	15 (9, 9)	3
1057	C	200	$U \sim [5, 20]$	114(73)	143(68)	257	15 (9, 12)	6
1058	C	300	$U \sim [25, 40]$	138(54)	183(62)	321	15 (9, 9)	3
1059	C	400	$U \sim [25, 40]$	196(40)	116(36)	312	15 (11, 6)	2
1060	C	500	$U \sim [25, 50]$	295(115)	353(157)	648	15 (9, 10)	4
1061	C	100	$U \sim [5, 20]$	73(22)	181(29)	254	20 (7, 15)	2
1062	C	200	$U \sim [5, 20]$	114(55)	201(92)	315	20 (11, 15)	6
1063	C	300	$U \sim [25, 40]$	201(83)	211(76)	412	20 (12, 13)	5
1064	C	400	$U \sim [25, 40]$	233(82)	176(86)	409	20 (14, 11)	5
1065	C	500	$U \sim [25, 50]$	314(155)	479(142)	793	20 (10, 15)	5
1066	C	100	$U \sim [5, 20]$	193(86)	236(103)	429	25 (16, 16)	7
1067	C	200	$U \sim [5, 20]$	247(87)	174(111)	421	25 (19, 14)	8
1068	C	300	$U \sim [25, 40]$	236(78)	288(88)	524	25 (14, 16)	5
1069	C	400	$U \sim [25, 40]$	297(144)	254(141)	551	25 (18, 15)	8
1070	C	500	$U \sim [25, 50]$	572(158)	355(113)	927	25 (17, 13)	5
1071	C	100	$U \sim [5, 20]$	268(54)	158(45)	426	30 (21, 13)	4
1072	C	200	$U \sim [5, 20]$	301(106)	176(106)	477	30 (25, 14)	9
1073	C	300	$U \sim [25, 40]$	348(123)	264(114)	612	30 (21, 16)	7
1074	C	400	$U \sim [25, 40]$	490(230)	298(210)	788	30 (25, 17)	12
1075	C	500	$U \sim [25, 50]$	704(277)	587(261)	1291	30 (20, 18)	8
1076	C	100	$U \sim [5, 35]$	20(0)	224(0)	244	10 (1, 9)	0
1077	C	200	$U \sim [5, 35]$	109(25)	107(32)	216	10 (7, 5)	2
1078	C	300	$U \sim [25, 50]$	245(154)	281(144)	526	10 (6, 8)	4
1079	C	400	$U \sim [25, 50]$	315(91)	147(118)	462	10 (9, 4)	3
1080	C	500	$U \sim [25, 75]$	289(42)	271(53)	560	10 (6, 5)	1
1081	C	100	$U \sim [5, 35]$	180(54)	204(20)	384	15 (7, 10)	2
1082	C	200	$U \sim [5, 35]$	119(8)	216(20)	335	15 (7, 9)	1
1083	C	300	$U \sim [25, 50]$	356(115)	309(120)	665	15 (10, 8)	3
1084	C	400	$U \sim [25, 50]$	356(40)	246(30)	602	15 (9, 7)	1
1085	C	500	$U \sim [25, 75]$	390(96)	468(113)	858	15 (7, 10)	2
1086	C	100	$U \sim [5, 35]$	278(78)	220(73)	498	20 (14, 9)	3
1087	C	200	$U \sim [5, 35]$	285(107)	286(108)	571	20 (12, 13)	5
1088	C	300	$U \sim [25, 50]$	566(121)	321(112)	887	20 (14, 9)	3
1089	C	400	$U \sim [25, 50]$	440(139)	432(167)	872	20 (12, 12)	4
1090	C	500	$U \sim [25, 75]$	652(159)	577(189)	1229	20 (13, 11)	4
1091	C	100	$U \sim [5, 35]$	404(172)	236(148)	640	25 (19, 15)	9
1092	C	200	$U \sim [5, 35]$	246(77)	360(90)	606	25 (14, 15)	4
1093	C	300	$U \sim [25, 50]$	575(307)	685(310)	1260	25 (15, 18)	8
1094	C	400	$U \sim [25, 50]$	542(133)	562(125)	1104	25 (14, 15)	4
1095	C	500	$U \sim [25, 75]$	918(281)	624(277)	1542	25 (19, 11)	5
1096	C	100	$U \sim [5, 35]$	453(116)	332(106)	785	30 (20, 16)	6
1097	C	200	$U \sim [5, 35]$	367(174)	385(175)	752	30 (19, 21)	10
1098	C	300	$U \sim [25, 50]$	578(343)	904(390)	1482	30 (15, 24)	9
1099	C	400	$U \sim [25, 50]$	626(314)	756(310)	1382	30 (18, 21)	9
1100	C	500	$U \sim [25, 75]$	936(453)	972(475)	1908	30 (20, 19)	9

Table 12: Data summary of set $S2_C$

References

[1] Adenso-Díaz, B., Lozano, S., Moreno, P., 2014. Analysis of the synergies of merging multi-company transportation needs. Transportmetrica A: Transport Science 10, 533–547.

[2] Agarwal, R., Ergun, Ö., 2010. Network design and allocation mechanisms for carrier alliances in liner shipping. Operations Research 58, 1726–1742.

[3] Ahr, D., 2004. Contributions to multiple postmen problems. PhD dissertation, Departament of Computer Science, Heidelberg University, Heidelberg, Germany .

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

[5] Archetti, C., Bianchessi, N., Irnich, S., Speranza, M.G., 2014. Formulations for an inventory routing problem. International Transactions in Operational Research 21, 353–374.

[6] Archetti, C., Hertz, A., Speranza, M.G., 2006. Metaheuristics for the team orienteering problem. Journal of Heuristics 13, 49–76.

[7] Archetti, C., Jabali, O., Speranza, M.G., 2015. Multi-period vehicle routing problem with due dates. Computers and Operations Research 61, 122–134.

[8] Archetti, C., F.E.H.M.n.D., 2017. The flexible periodic vehicle routing problem. Computers & Operations research To appear.

[9] Audy, J.F., Lehoux, N., D'Amours, S., Rönnqvist, M., 2012. A framework for an efficient implementation of logistics collaborations. International Transactions in Operational Research 19, 633–657.

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

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

[12] Chao, I.M., Golden, B.L., Wasil, E., 1993. A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. American Journal of Mathematical and Management Sciences 13, 371–406.

[13] Corberán, A., Fernández, E., Franquesa, C., Sanchis, J., 2011. The windy clustered prize-collecting arc routing problem. Transportation Science 45, 317–344.

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

[15] Cruijssen, F., Braysy, O., Dullaert, W., Fleuren, H., Salomon, M., 2007a. Joint route planning under varying market conditions. International Journal of Physical Distribution and Logistics Management 37, 287–304.

[16] Cruijssen, F., Cools, M., Dullaert, W., 2007b. Horizontal cooperation in logistics: Opportunities and impediments. Transportation Research Part E: Logistics and Transportation Review 43, 129 –142.

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

[18] Dai, B., Chen, H., 2011. A multi-agent and auction-based framework and approach for carrier collaboration. Logistics Research 3, 101–120.

[19] Defryn, C., Sörensen, K., Cornelissens, T., 2016. The selective vehicle routing problem in a collaborative environment. European Journal of Operational Research 250, 400–411.

[20] Ergun, Ö., Kuyzu, G., Savelsbergh, M., 2007a. Reducing truckload transportation costs through collaboration. Transportation Science 41, 206–221.

[21] Ergun, Ö., Kuyzu, G., Savelsbergh, M., 2007b. Shipper collaboration. Computers and Operations Research 34, 1551–1560. Part Special Issue: Odysseus 2003 Second International Workshop on Freight Transportation Logistics.

[22] Fernández, E., Fontana, D., Speranza, M.G., 2016. On the collaboration uncapacitated arc routing problem. Computers and Operations Research 67, 120–131.

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

[24] Fisher, M.L., J.R., 1981. A generalized assignment heuristic for vehicle routing. Networks 11, 109–214.

[25] Ford, L., Fulkerson, D., 1962. Flows in Networks. Princeton University Press, Princeton, N.J.

[26] Gillett, B.E., Miller, L.R., 1974. A heuristic algorithm for the vehicle-dispatch problem. Operations Research 22, 340–349.

[27] Golden, B.L., Magnanti, T.L., Nguyen, H.Q., 1977. Implementing vehicle routing algorithms. Networks 7, 113–148.

[28] Gouveia, L., 1995. A result on projection for the vehicle routing ptoblem. European Journal of Operational Research 85, 610–624.

[29] Hopcroft, J., R.E., T., 1973. Efficient algorithms for graph manipulation. Communications of the ACM 16, 372–378.

[30] Karsten, F., Slikker, M., van Houtum, G.J., 2015. Resource pooling and cost allocation among independent service providers. Operations Research 63, 476–488.

[31] Krajewska, M., Kopfer, H., 2006. Collaborating freight forwarding enterprises. OR Spectrum 28, 301–317.

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

[33] Kuyzu, G., Akyol, C.G., Ergun, Ö., Savelsbergh, M., 2015. Bid price optimization for truckload carriers in simultaneous transportation procurement auctions. Transportation Research Part B: Methodological 73, 34–58.

[34] Letchford, A.N., Salazar-González, J.J., 2015. Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. European Journal of Operational Research 244, 730–738.

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

[36] Nadef, D., Rinaldi, G., 2014. Branch-and-cut algorithms for the capacitated vrp, in: Toth, P., Vigo, D. (Eds.), The Vehicle Routing Problem. Society for Industrial and Applied Mathematics.

[37] Nemhauser, G., Wolsey, L., 1988. Applications of Special-Purpose Algorithms. John Wiley and Sons, Inc.

[38] Özener, O., Ergun, Ö., Savelsbergh, M., 2011. Lane-exchange mechanisms for truckload carrier collaboration. Transportation Science 45, 1–17.

[39] Pomponi, F., Fratocchi, L., Tafuri, S., Palumbo, M., 2013. Horizontal collaboration in logistics: a comprehensive framework. Research in Logistics and Production 3, 243–254.

[40] Sprenger, R., Mönch, L., 2012. A methodology to solve large-scale cooperative transportation planning problems. European Journal of Operational Research 223, 626–636.

[41] Tillman, F., 1969. The multiple terminal delivery problem with probabilistic demands. Transportation Science 3, 192–204.

[42] Tillman, F., Cain, T., 1972. An upper bound algorithm for the single and multiple terminal delivery problem. Management Science 18, 664–82.

[43] Toth, P., Vigo, D., 2014. Vehicle Routing. Society for Industrial and Applied Mathematics, Philadelphia, PA.

[44] Vanovermeire, C., Sörensen, K., Breedam, A.V., Vannieuwenhuyse, B., Verstrepen, S., 2014. Horizontal logistics collaboration: decreasing costs through flexibility and an adequate cost allocation strategy. International Journal of Logistics Research and Applications 17, 339–355.

[45] Verdonck, L., Caris, A., Ramaekers, K., Janssens, G.K., 2013. Collaborative logistics from the perspective of road transportation companies. Transport Reviews 33, 700–719.

[46] Vornhusen, B., Wang, X., Kopfer, H., 2014. Vehicle routing under consideration of transhipment in horizontal coalitions of freight carriers. Procedia CIRP 19, 117–122. 2nd CIRP Robust Manufacturing Conference (RoMac 2014).

[47] Wang, X., Kopfer, H., 2014. Collaborative transportation planning of less-than-truckload freight. OR Spectrum 36, 357–380.

[48] Wren, A., Holliday, A., 1972. Computer scheduling of vehicles from one or more depots to a number of delivery points. Operational Research Quarterly 23, 333–344.