I&EC research
Industrial & Engineering Chemistry Research

Article

pubs.acs.org/IECR

A Branch-and-Price Approach To Manage Cargo Consolidation and Distribution in Supply Chains

Rodolfo G. Dondo* and Carlos A. Mendez

INTEC (Universidad Nacional del Litoral—CONICET), Güemes 3450, (3000) Santa Fe, Argentina

Supporting Information

ABSTRACT: The optimization of an enterprise supply chain requires reducing costs and inventories. One usual way to increase the efficiency of the supply chain is to outsource the movement of shipments on third-party logistics (3PL) companies. In turn, 3PL companies are required to consolidate shipments from different suppliers in the outbound vehicles at a terminal of the carrier terminals-network. Services to manufacturing enterprises are differentiated into two groups, with regard to the shipment size. Manufacturers contract full-truckload (FTL) carriers when a shipment is large enough to fill a vehicle whereas less-than-truckload (LTL) carriers are used for minor shipments. LTL carriers must convey several transportation-requests from the origin locations to their destinations while minimizing the total movements-cost by using the possibility of goods-transshipments on the carrier's terminals-network. We present a methodology for finding near-optimal solutions to a problem related to LTL-shipping by using column generation combined with a customized branch-and-price procedure. The approach rapidly provides near-optimal solutions, since it solves the column generation subproblems approximately and does not necessarily consider all unexplored nodes in the search-tree. We also present computational results on numerous test problems of varied topologies and on a real case study.

1. INTRODUCTION

Chemical and industrial companies usually perform a series of activities, such as purchasing raw materials from suppliers, manufacturing and storing end-products at intermediate facilities, and delivering them to final customers. Suppliers, manufacturers, warehouses, and customers are the major components of the so-called supply chain (SC), carrying goods from the upstream side to the downstream side of the SC.[1] Distribution is concerned with the shipment and storage of products downstream from the supplier side to the customers side in the SC.[2]

For companies in the process industry to remain competitive and economically viable, this requires, for the value preservation part of the industry, optimization of the enterprise and its supply chain by reducing costs and inventories, operating efficiently, and continuously improving product quality.[3] Therefore, good coordination of the SC is a critical issue in most manufacturing companies. Supply chain management (SCM) aims to efficiently control the material flow through the SC. One usual way to increase the efficiency of the SC is to outsource the movement of shipments on third-party logistics (3PL) companies that operate with a very high level of efficiency. Small-scale manufacturing companies usually lack the resources to develop their own logistics leg and therefore are forced to outsource. In those cases, 3PL companies are required to consolidate some shipments from different suppliers in the outbound vehicles at a terminal of the carrier terminals-network. 3PL companies provide transportation services to manufacturing enterprises for shipping freight, and such companies are differentiated into two groups, with regard to the size of the shipment. Manufacturers contract full-truckload (FTL) carriers when a shipment is large enough to fill a vehicle capacity, whereas less-than-truckload (LTL) carriers are used for minor shipments. LTL transportation typically involve shipments ranging from 70 kg to 10 000 kg. A typical shipment occupies only 5%–10% of the trailer capacity.[4] How freight is routed through the terminals-network, and thus where opportunities for consolidation occur, is determined by the so-called "load plan", which specifies, if convenient, a sequence of transfers for each shipment.[4] The carriers usually operate in multiechelon networks and the modality includes express carriers that provide urgent, time-definite services as well as carriers that supply flexible-time services for small industrial shipments. In the lower-level of the network, the local routes visit customers to pick up or deliver goods and the upper level is a long-haul subnetwork that connects the main distribution centers. Sometimes, a subset of distribution centers or hubs may exist at a higher hierarchical level. There, the inbound freight of long-haul routes can be sorted and shipped to other distribution centers. In order to operate with high efficiency, an LTL system must deal with complex issues, such as how truck loading and unloading tasks should be scheduled at the terminals and how vehicles should be routed to collect and deliver cargo. The way goods are collected and delivered is of crucial importance for determining the cargo flows and workload on the terminals and the following shipping alternatives to move requests from the manufacturer-locations to the destinations may be considered:

Special Issue: Jaime Cerdá Festschrift

Received: January 15, 2014
Revised: March 19, 2014
Accepted: March 26, 2014
Published: March 26, 2014

(i) Shipping directly from the origin to the destination location.
(ii) Shipping from the origin to the destination after crossdocking in a terminal.
(iii) Shipping from the origin to the destination through a long-haul trip between the two terminals where the load is cross-docked.
(iv) There is no limit on the number of stops on the terminals.

However, cost-effective shipping is not the only challenge for carriers since they have to ensure a certain service-quality level in terms of delivery time and frequency of service. Clearly, the programming of the daily operational shipping plan is a very challenging task.

The present work presents a truncated branch-and-price decomposition-approach to provide solutions to a problem related to the LTL shipping-mode. The problem arises from a small transportation company that provides conveyance services to manufacturing and retailer companies.

The remainder of the paper is organized as follows. In section 2, we review the literature on LTL-shipping and on the numerical technique used to solve the studied problem. The problem is first described and later formulated as an integer program in section 3. A decomposition algorithm based in the branch-and-price paradigm to solve the problem is detailed in section 4. Some computational results and a solution to a real case study are presented in section 5. Section 6 presents the conclusions of the work.

2. LITERATURE REVIEW

From a supply chain (SC) perspective, Amaro and Barvosa-Povoa[5] presented a discrete mixed-integer linear programming (MILP) formulation to provide the optimal scheduling of SC networks. The model provides a detailed operational plan at the production, storage, and transportation levels. Dondo et al.[2] presented an optimizing approach to the short-term operational planning of multiechelon transportation networks. Shipping of commodities from factories to customers directly and/or via distribution centers were considered. However, now, a 3PL perspective aimed at integrating the efficient operation of several supply chains was taken into account. In the other hand, load plan design and scheduling for LTL shipping leads to challenging optimization problems. Because of the intrinsic NP-hardness of most routing problems included in the design of the plan, the mathematical programming approach was not considered a feasible choice for solving and coordinating them until a few years ago. Therefore, early research focused on relatively simple local search heuristics for models based on static networks that do not consider timing constraints. For example, Powell and co-workers[6−9] studied the load planning problem, which is defined as the specification of how freight should be routed and consolidated over a network defined by direct services between terminals. The authors proposed several heuristic procedures based on the hierarchical decomposition of the problem into a master problem and several slave subproblems. The master problem is a network design problem in which direct services offered by the carrier are established and a minimum service frequency is imposed. The total system cost is computed for each configuration of the selected services. In the work of Farvolden and Powell,[10] a dynamic model was presented to more accurately describe the consolidation operations. A more advanced heuristic procedure was also developed. It relies on determining service network arc subgradients by solving large-scale multicommodity network flow problems. The approach allows freight for a specific origin-destination request to be moved over multiple paths. Relatively little analysis has been done about long-haul network shipments from a vehicle routing perspective. To the best of our knowledge, this approach was introduced by Kuby and Gray,[11] and Lin and Chen[12] later extended it to the hierarchical hub-and-spoke network design problem for LTL carriers. In these works, stop-over strategies were considered to increase the load associated with the flow between terminals. Estrada and Robusté[13] relied on a metaheuristic approach and then developed a tabu-search algorithm to solve the long-haul routing design problem with capacitated distribution centers and time-constrained shipments. The method uses direct, hub-and-spoke, and stop-over strategies to allocate shipments to routes. Barcos et al.[14] relied on an ant-colony metaheuristic procedure to solve an LTL problem, because the mathematical formulation used to model it needed a large number of variables and constraints. Jarrah et al.[15] presented a mathematical formulation for the service network design problem in the context of large-scale LTL freight operations. The formulation fragments a massive network model with up to 1.3 million binary variables and 1.3 million rows into an efficient integer programming problem and a coordinating master network-design problem. The authors claim that they were able to produce high-quality solutions within very reasonable CPU times. Toptal et al.[16] studied the transportation pricing problem of a truckload carrier consisting of a retailer, a truckload carrier, and an LTL carrier. In this setting, the truckload carrier makes the pricing decision based on previous knowledge on the LTL's price schedule and the retailer's ordering behavior. Erera et al.[4] presented a sophisticated computational approach for the tactical load plan used by an LTL carrier. The generated load plan determines how freight is routed through an LTL carrier's line-haul terminal-network by specifying a sequence of transfer terminals for all freight shipments without a time schedule of trailer, tractor, and driver dispatches. Given the load plan, a scheduling approach creates the truck dispatches between terminals and then creates cyclic driver schedules to cover all dispatches. In the field of vehicle routing, many authors have devised formulations and solution approaches based on branch-and-price for increasingly complex variations of the capacitated vehicle routing problem. (See refs 17−19 for discussion about some of the methodologies.) Since transportation companies usually have more than one depot and they often manage a fleet of vehicles with different capacities and operating costs, Bettinelli et al.[20] developed a branch-cut-price solution methodology for the multidepot vehicle routing problem with time windows (MD-VRPTW). Santos et al. applied that methodology first to the vehicle routing problem with cross-docking (VRPCD)[21] and later to the so-called pick-up and delivery problem with crossdocking (PDPCD).[22] In this work, we unify and generalize the approaches from refs 20 and 22 to define a problem arising from a 3PL company that is required to consolidate shipments from different suppliers in the outbound vehicles at some terminal of the company terminals-network. Also, we propose an efficient column generation (CG)-based branch-and-price procedure to solve the problem.

3. MODELING AND DEFINING THE PROBLEM

An LTL carrier operates a terminals-network to provide conveyance services during a specified time period as, for example, on a daily basis. The modality include many practical variations but the system usually operates as follows: during a given time horizon, "local carriers" pick up shipments from various source locations in a given geographical area, and bring them to the terminal serving the area, which is usually called the "end-of-line" terminal. The terminal operates as a sorting and consolidation center and as a loading/unloading facility for the outbound and inbound freight of the area. After sorting and consolidation, large carriers are sent to other end-of-line terminals. Outbound freight from an end-of-line is sent to a "break-bulk" terminal, where it may be consolidated with freight from other end-of-line terminals. The terminals-network of the carrier and the cargo-source and destination locations to visit are illustrated in Figure 1. By consolidating freight from

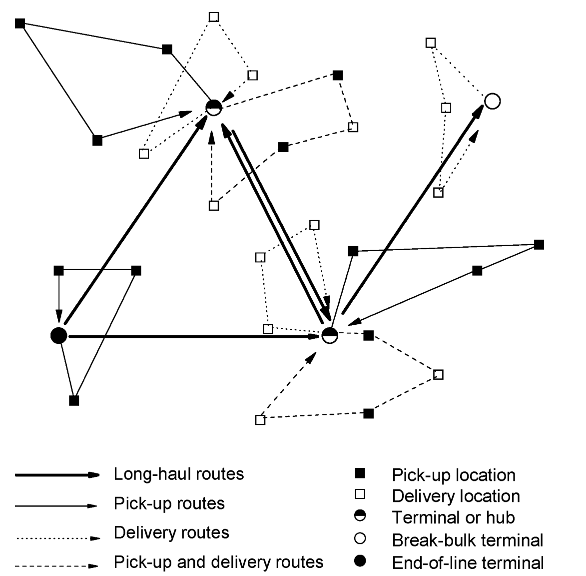

Figure 1. A typical LTL carrier system, using different delivery options in a two-level network.

and destined to several locations, break-bulk terminals handle freight to dispatch cost-effective trailer loads. Usually long-haul carriers are different from local carriers. The former may be large trucks or trains while the latter are smaller, "urban" trucks. Consolidation at a terminal requires freight to be cross-docked, which results in handling costs. Furthermore, a shipment may be delivered to its destination via several options:

(i) it can be picked up and delivered by the same local carrier if the source and destination places are located nearby;
(ii) it may be sent to the terminal serving the area and, from there, delivered to the destination; and
(iii) from one terminal, it may be sent to another terminal and, from there, delivered to the destination.

The delivery agenda defines how freight is routed by specifying a sequence of transfer operations for each shipment. It also determines the paths of the shipments, as well as the local routes and long-haul trips. The particular problem studied here is formally defined as follows.

The transportation-network is represented by a directed graph $G(T \cup I^{+} \cup I^{-}; A)$ comprising a set T of terminals that operate as the origin and destination of local and long-haul shipments; it also includes a set I^{+} of pick-up locations and a set I^{-} of delivery sites. The list of route-arcs is defined by A. Non-negative values d_{ij} and t_{ij} are associated with each arc $(i,j) \in A$, respectively representing the travel distance/cost and the travel-time necessary to reach site j starting from location i. A transportation request $\tau = \{i,j\}$ of a request list $\Gamma = \{\tau_1, ..., \tau_n\}$ consists of a demand for a transportation service from the origin-location $i \in \{I^{+} \cap \tau\}$ to destination location $j \in \{I^{-} \cap \tau\}$ for a stated load l_{ij}. Visits must start within stated time windows $[t_i^{\min}, t_i^{\max}]$ for all pick-up sites $i \in I^{+}$ and $[t_j^{\min}, t_j^{\max}]$ for all delivery sites $j \in I^{-}$. These time-windows must also be compatible, so they must not generate a solution infeasibility. Service times st_i at each pick-up/delivery location $i \in \{I^{+} \cup I^{-}\}$ have two components: a fixed preparation time t_f and a variable component that depends on the size of the load to pick-up/deliver t_v. Thus, $\mathrm{st}_i = t_f + t_v |l_i|$. The following shipping alternatives are available to fulfill the delivery of any request $\tau \in \Gamma$:

(i) (i) Shipping on a local vehicle directly from origin $i \in \{I^{+} \cap \tau\}$ to destination $j \in \{I^{-} \cap \tau\}$.
(ii) Shipping from origin $i \in \{I^{+} \cap \tau\}$ to destination $j \in \{I^{-} \cap \tau\}$ via cross-docking on a single terminal $t \in T$.
(iii) Shipping from origin $i \in \{I^{+} \cap \tau\}$ to destination $j \in \{I^{-} \cap \tau\}$ through a long-haul trip between two terminals $(t,t') \in T$: $t \neq t'$.

The number of trips of any type, the terminals from where local trips start/end, and the long-haul flow between terminals must be determined by the problem solution. The operational costs are dependent on (i) the number of local and long-haul routes and (ii) the number of incurred cross-docking operations. The objective is to minimize the sum of cross-docking costs, vehicle fixed costs, and vehicle traveling costs while satisfying the following operational constraints:

- All pick-up and delivery sites must be visited just once and by only one vehicle.
- The service at each customer must start within its time window.
- Each pick-up/delivery/mixed route begins at a terminal and ends at the same terminal.
- The sum of the collected/delivered loads in each pick-up/delivery/mixed route must not exceed the capacity of the in-route vehicle.
- Pickup routes must be fulfilled within the time interval $[0,t^{\max(+)}]$.
- The delivery routes must be fulfilled within the time interval $[t^{\min(-)},t^{\max(-)}]$.
- The mixed pick-up and delivery routes must be fulfilled within the time interval $[0,t^{\max(-)}]$.
- Long-haul trips between terminals are carried out within the time interval $[t^{\max(+)},t^{\min(-)}]$.

In order to model this problem as an integer program (IP), let us denote R^{T} as the set of transfers (i.e., long-haul routes and cross-docking operations in a single terminal), R^{+} the set of feasible pick-up routes, R^{-} the set of feasible delivery routes, and R^{+-} the set of feasible mixed pick-up and delivery routes. For each route $r \in \{R^{\mathrm{T}} \cup R^{+} \cup R^{-} \cup R^{+-}\}$, c_r denotes its cost, given by the sum of the costs of the arcs traveled by the vehicle plus a given fixed vehicle-utilization-cost. Transfers $r \in R^{\mathrm{T}}$ also include the cost of the associated cross-docking operations at start/end terminals. Assume that we are given a binary parameter a_{ir}, to indicate whether route $r \in \{R^{+} \cup R^{-} \cup$

$R^{+-}\}$ does or does not visit location $i \in \{I^+ \cup I^-\}$: if it does, $a_{ir} = 1$; if it does not, $a_{ir} = 0$. For a route $r \in \{R^+ \cup R^- \cup R^{+-}\}$, one must also consider a binary parameter b_{rt} that assumes a value of 1 if route r starts/ends on terminal t or, otherwise, assumes a value of 0. In that model, we use the binary decision variable X_r to determine whether route $r \in \{R^T \cup R^+ \cup R^- \cup R^{+-}\}$ does or does not belong to the optimal solution. The problem can now be formulated as

$$\text{minimize} \sum_{r \in R^T} c_r X_r + \sum_{r \in R^+} c_r X_r + \sum_{r \in R^-} c_r X_r + \sum_{r \in R^{+-}} c_r X_r \tag{1}$$

subject to

$$\sum_{r \in R^{+-}} a_{ir} X_r + \sum_{r \in R^+} a_{ir} X_r = 1 \qquad \forall\, i \in I^+ \tag{2}$$

$$\sum_{r \in R^{+-}} a_{ir} X_r + \sum_{r \in R^-} a_{ir} X_r = 1 \qquad \forall\, i \in I^- \tag{3}$$

$$\sum_{t \in T} b_{tr}^{\text{start}} \sum_{r \in R^+} a_{ir} b_{rt} X_r + \sum_{t \in T} b_{tr}^{\text{end}} \sum_{r \in R^-} a_{ir} b_{rt} X_r - 1 \le X_r$$
$$\forall\, r \in R^T, \tau = \{i, j\} \in \Gamma\colon i \in I^+ \cap \tau, j \in I^- \cap \tau, X_r = \{0, 1\} \tag{4}$$

The objective function described by expression 1 minimizes the total system cost, i.e., the cost of all types of routes. Constraint (2) ensures that the source site $i \in I^+$ is visited exactly once, while constraint (3) guarantees that each destination place $i \in I^-$ is visited exactly once. The inequality described by expression 4 represents long-haul-transfer constraints, imposing that the long-haul route $r = (t,t') \in R^T$ is used whenever the request-load picked up from source-site $i \in \{I^+ \cap \tau\}$ is unloaded on terminal t and loaded on terminal t' for its final delivery to destination $i \in \{I^- \cap \tau\}$. So, if a given transportation request is moved from its pick-up site to terminal t by route $r \in R^+$ and later is delivered to its destination from the terminal t' by route $r \in R^-$, then it must travel from t to t' via the long-haul route $r = (t,t') \in R^T$ that starts on t and ends on t'. Both indexes t and t' may refer to the same physical terminal to consider shipping option (ii). So, this IP formulation allows shipping modes (i), (ii), and (iii) to be modeled as described by the problem definition. Since the number of terminals is much smaller than the number of pick-up and delivery locations, and because the transfer routes involve the use of a single arc, transfers can be totally enumerated. For example, an instance comprising 4 terminals will lead to a problem with, at most, $4^2 = 16$ possible transfers (12 long-haul routes plus 4 pure cross-docking operations on a single terminal). Note that side constraints as capacity and time constraints are not considered explicitly here. They are considered implicitly, because the formulation only accounts for legal routes (i.e., routes that fulfill these side constraints).

4. THE SOLUTION ALGORITHM

In this section, the model defined by expressions 1−4 is embedded into a branch-and-price procedure in order to generate solutions for the problem formulated above. The formulation represents the set of all feasible routes and its objective is to select the minimum-cost subset of routes such that each transportation request is fulfilled, *no matter how*. It is not possible to generate all feasible routes but the CG approach handles this complexity by implicitly considering all of them through the solution of the linear relaxation of the formulation described by expressions 1−4, called the reduced master problem (RMP). In this way, a portion of feasible routes (usually an initial but suboptimal solution) is enumerated and the linear relaxation of the RMP is solved by considering only this partial set. The solution to this problem is used to determine if there are routes not included in the routes-set that can reduce the objective function value. Using the values of the optimal dual variables for the master constraints, with respect to the partial routes-set, new routes are generated and incorporated into the columns pool, and the linear relaxation of the RMP is solved again. The procedure iterates between the master problem and the slave route−generator problem(s) until no routes with negative reduced costs can be found. Finally, a master IP may be solved to determine the best subset of routes. Although the solution found may not be the global optimal, it is generally "close". To determine the optimal subset, the procedure must be embedded into a branch-and-bound algorithm, because some routes that were not generated when solving the relaxed RMP may be needed to solve the IP.

In summary, the procedure starts with an initial feasible solution and decomposes the problem into a master−slave structure comprising the relaxed RMP and the slave tour-generator problem(s). The master−slave structure is recursively solved until no feasible routes can be generated. In that case, the RMP is solved again to verify the solution integrality. If the solution to this problem is an integer, the optimal solution to the LTL problem has been found and the procedure ends. Otherwise, the integer solution to the RMP (or the global upper bound, GUB) will have a value higher than the value of the solution to the relaxed RMP (or the global lower bound, GLB). Therefore, the procedure starts branching to generate the missing routes. At each tree-node, the mechanism is repeated and the bounds are compared. If the local lower bound (LLB), given by the value of the relaxed RMP, is higher than the GUB (given by the best available integer solution), the node is fathomed; otherwise, it is divided into two child-nodes that are included in the database of unsolved subspaces. Afterward, the next subspace is fetched from the database until this base is empty. Finally, the solution is specified by solving, for each selected column, a traveling salesman problem with time windows. The algorithm is sketched in Figure 2.

The entire process is called *branch-and-price* and involves the definition of the linear RMP, the definition of the *slave problem(s)* or *pricing problem(s)*, and the implementation of a *branching rule*.

4.1. The Master Problem.

The IP formulation described by expressions 1−4 contains several binary variables, the number of which grows with the size of the set of feasible routes. In order to compute a lower bound to its objective function value, we relax the integrality condition for variables X_r and consider the IP to be the RMP. Initially, the problem includes a few columns, representing a feasible solution, and the cost of these columns is known in advance. The usual decomposition approach for vehicle routing similar problems cannot be applied to this LTL-type problem, because not only partitioning constraints (2) and (3) but also transfer constraint (4) must go into the master program. These are linking constraints ensuring that a given cargo must be moved through the arc connecting the used terminals. So, if a given transportation request is moved from its pick-up site to terminal t by a route $r \in R^+$ (i.e., $\sum_{t \in T} b_t^{\text{start}} \sum_{r \in R^+} a_{ir}\, b_{rt}\, x_r = 1$) and later is delivered to its destination j from terminal t' by a route $r \in R^-$ (i.e., $\sum_{t \in T} b_t^{\text{end}} \sum_{r \in R^-} a_{jr}\, b_{rt'}\, x_r = 1$), then it must

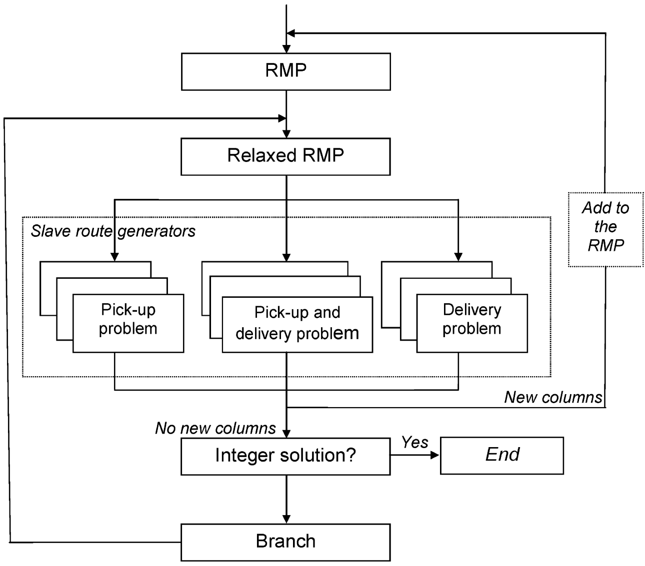

Figure 2. Outline of the branch-and-price algorithm.

travel from terminal t to terminal t' via the long-haul route, $r = (t,t') \in R^T$. Since CG is applied to master problems with the following structure:[19]

$$\text{minimize} \sum_{r\in R} c_r X_r$$

subject to

$$\sum_{r\in R} \mathbf{a}_r X_r \leq \mathbf{b}$$

We just need to reorder constraint (4) to give rise to the following relaxed RMP:

$$\text{minimize} \sum_{r\in R^T} c_r X_r + \sum_{r\in R^+} c_r X_r + \sum_{r\in R^-} c_r X_r + \sum_{r\in R^{+-}} c_r X_r \tag{1'}$$

subject to

$$\sum_{r\in R^+} a_{ir} X_r + 0 + \sum_{r\in R^{+-}} a_{ir} X_r + 0 \geq 1 \qquad \forall\, i \in I^+ \tag{2'}$$

$$0 + \sum_{r\in R^-} a_{ir} X_r + \sum_{r\in R^{+-}} a_{ir} X_r + 0 \geq 1 \qquad \forall\, i \in I^- \tag{3'}$$

$$\sum_{r\in R^+} \alpha_{ir}^+ X_r + \sum_{r\in R^-} \alpha_{ir}^- X_r + 0 - x_r \leq 1$$
$$\forall\, r \in R^T,\ \tau = \{i, j\} \in \Gamma\colon i \in I^+ \cap \tau,\ j \in I^- \cap \tau,\ 0 \leq X_r \leq 1 \tag{4'}$$

The binary parameters α_{ir}^+ and α_{ir}^- are defined as follows:

$$\alpha_{ir}^+ = \sum_{t\in T} b_t^{\text{start}} b_{rt} a_{ir} \qquad \forall\, r \in R^T,\ i \in (I^+ \cap r) \tag{5a}$$

$$\alpha_{ir}^- = \sum_{t\in T} b_t^{\text{end}} b_{rt} a_{ir} \qquad \forall\, r \in R^T,\ i \in (I^- \cap r) \tag{5b}$$

Routing problems are naturally modeled by a partitioning formulation, because each customer must be visited just once but they can also be formulated as a set covering problem in which each customer must be visited at least once. Since visiting a location once results in the less costly feasible subset, an optimal set-covering solution will also be an optimal set-partitioning solution. When there is a choice between a partitioning and a covering formulation, the last one is usually preferred, since it is numerically more stable and easier to solve.[17] Now, let us assume that the optimal solution to the relaxed RMP had been found and that π^+, π^-, and π^t are the vectors of optimal dual variables values for constraints (2), (3), and (4), respectively. These vectors are to be passed to the slave pricing problems in order to produce more routes that will be useful to later reduce the value of the objective function described by expression 1. Therefore, pricing problems aimed at minimizing the following three objectives must be solved:

$$\begin{aligned}\hat{c}_r^+ &= c_r - \sum_{i\in I^+} \pi_i^+ a_{ir} - \sum_{r\in R^T}\sum_{i\in I^+} \pi_{ri}^{t} \alpha_{ir}^+ \\ &= c_r - \sum_{i\in I^+} \pi_i^+ a_{ir} - \sum_{r\in R^T}\sum_{t\in T} b_{tr}^{\text{start}} \sum_{i\in I^+} \pi_{ri}^{t} b_{rt} a_{ir}\end{aligned} \tag{6a}$$

$$\begin{aligned}\hat{c}_r^- &= c_r - \sum_{i\in I^-} \pi_i^- a_{ir} - \sum_{r\in R^T}\sum_{i\in I^-} \pi_{ri}^{t} \alpha_{ir}^- \\ &= c_r - \sum_{i\in I^-} \pi_i^- a_{ir} - \sum_{r\in R^T}\sum_{t\in T} b_{tr}^{\text{end}} \sum_{i\in I^-} \pi_{ri}^{t} b_{rt} a_{ir}\end{aligned} \tag{6b}$$

$$\hat{c}_r^{+-} = c_r - \sum_{i\in I^+} \pi_i a_{ir}^{+} - \sum_{i\in I^-} \pi_i a_{ir}^{-} \tag{6c}$$

The three objective functions define the respective slave problems, one for each route type. In this way, eq 6a is the objective for the search of new pick-up routes, eq 6b is the objective for finding new delivery routes, and eq 6c is the objective for finding new mixed routes. At each CG iteration, the linear relaxation of the RMP is first solved and then we search for new columns with a negative reduced cost by solving the corresponding pricing problems.

4.2. The Pricing Problems. Branch-and-price approaches for vehicle routing problems use route selection variables in the master program, and there is one binary variable for each route. Each feasible tour is an elementary path from a start-terminal to the same end-terminal through some locations of the network. The pricing problems are elementary shortest path problems with resource constraints (ESPPRCs) and when there are multiple terminals, a pricing problem may be solved for each terminal in each pricing step. The pricing problems are NP-hard in the strong sense, and there is a considerable amount of literature about the solution of ESPPRCs. (See ref 23 as an introductory work.) The most commonly used technique to solve the pricing problems is dynamic programming, in which a relaxation of the pricing algorithm is solved and cycles are allowed. Nevertheless, the recent trend relies on dynamic programming, in which pricing problems are solved exactly without allowing cycles. (See refs 24 and 26.) Algorithms that provide elementary routes may be further classified in dynamic programming procedures,[24,25] MILP formulations,[27] or constraint-programming algorithms.[28] In our application, we solve exactly the MILP formulation of the elementary pricing problems with a branch-and-cut solver. In summary, we enumerate transfers prior to the start of the master−slave recursion and encode the routing generator problems according to the MILP formulations that will be explained next.

4.2.1. The Slave Pick-Up Problem. The objective of the slave problem for designing pick-up columns is to find a route r that minimizes the quantity stated by eq 6d and is subject to the constraints stated by eqs 7−16.

$$\text{minimize}\left(\text{CV} - \sum_{i \in I^+} \pi_i Y_i - \sum_{r \in R^{\text{T}}} \sum_{t \in T} b_{tr}^{\text{start}} \sum_{i \in I^+} \pi_{ri}^t x_t Y_i\right) \tag{6d}$$

subject to

$$\sum_{t \in T} x_t = 1 \tag{7a}$$

$$x_t = 1 \qquad t = \text{selected terminal} \tag{7b}$$

$$D_i \geq \sum_{t \in T} x_t d_{ti} \qquad \forall\ i \in I^+ \tag{8}$$

$$\{D_j \geq D_i + d_{ij} - M_D(1 - S_{ij}) - M_D(2 - Y_i - Y_j)\} \quad \forall\ (i, j) \in I^+ : i < j \tag{9a}$$

$$\{D_i \geq D_j + d_{ij} - M_D S_{ij} - M_D(2 - Y_i - Y_j)\} \quad \forall\ (i, j) \in I^+ : i < j \tag{9b}$$

$$\text{CV} \geq \text{cf}_v + D_i + \sum_{t \in T} x_t d_{ti} - M_C(1 - Y_i) \qquad \forall\ i \in I^+ \tag{10}$$

$$T_i \geq \sum_{t \in T} x_t t_{ti} \qquad \forall\ i \in I^+ \tag{11}$$

$$\{T_j \geq T_i + \text{st}_i + t_{ij} - M_T(1 - S_{ij}) - M_T(2 - Y_i - Y_j)\} \quad \forall\ (i, j) \in I^+ : i < j \tag{12a}$$

$$\{T_i \geq T_j + \text{st}_j + t_{ji} - M_T S_{ij} - M_T(2 - Y_i - Y_j)\} \quad \forall\ (i, j) \in I^+ : i < j \tag{12b}$$

$$\text{TV} \geq T_i + \sum_{t \in T} x_t t_{ti} - M_T(1 - Y_i) \qquad \forall\ i \in I^+ \tag{13}$$

$$t_i^{\min} \leq T_i \leq t_i^{\max} \qquad \forall\ i \in I^+ \tag{14}$$

$$\text{TV} + \sum_{i \in I^+} Y_i \text{st}_i \leq t^{\max(+)} \tag{15}$$

$$\sum_{i \in I^+} Y_i l_i \leq q \tag{16}$$

The objective function described by expression 6d is the cost CV of the generated route minus the prices collected on the visited pick-up sites and the prices related to the inbound load-flow on the selected terminal. This equation is the pricing reformulation of eq 6a, where the parameter a_{ir} of the master problem becomes the decision variable Y_i of the pricing one. The binary parameter x_t indicates the start/end terminal of the designed tour. Constraint (8) set the minimum distance to reach site $i \in I^+$ as the distance of going directly from the terminal to location i. Constraints (9) and (10) compute the distances traveled to reach the visited sites $i \in I^+$ and the total cost of the generated route, respectively. So, eqs 9 fix the accumulated distance up to each visited site. If locations i and j are allocated onto the generated route ($Y_i = Y_j = 1$), the visiting ordering for both sites is determined by the value of the sequencing variable S_{ij}. If location i is visited before location j ($S_{ij} = 1$), according to constraints (9a) and (9b), the distance traveled to location j (D_j) must be larger than D_i by at least d_{ij}. In case node j is visited earlier ($S_{ij} = 0$), the reverse statement holds and constraint (9b) becomes active. If one or both sites are not allocated to the tour, constraints (9a) and (9b) become redundant. M_D is an upper bound for variables D_i. Equation 10 computes the route-cost CV via the addition of the fixed vehicle utilization cost cf_v to the traveled-distance-cost to the terminal to which the vehicle must return after visiting the last site. Since the last visited pick-up location cannot be known before the problem is resolved, eq 10 must be written for each site $i \in I^+$. M_C is an upper bound for the variable CV. The timing constraints stated by eqs 11–13 are similar to constraints (8)–(10), but they apply to the time dimension. M_T is an upper bound for the times T_i spent to reach the nodes $i \in I^+$ and for the tour-time-length TV. Equation 14 forces the service time on any site $i \in I^+$ to start at a time T_i bounded by the time window $[t_i^{\min}, t_i^{\max}]$. Equation 15 adds a term related to the unload activities on the selected terminal to the tour time-length, to define the end-time for all download activities. Equation 16 is a capacity constraint for the vehicle traveling the designed tour.

4.2.2. The Slave Delivery Problem. The objective of the slave problem for generating delivery tours is to find a route r that minimizes the quantity stated by the objective function described by 6e and subject to the constraints stated by eqs 17–26. The objective function is the pricing reformulation of eq 6b, and the constraints are similar to constraints 7–16, but they are used to design delivery routes. The difference, with respect to the former constraints, is that delivery tours must start after the start-time $t^{\text{start}(-)}$ and all activities must end before $t^{\max(-)}$. These differences are considered in eqs 21 and 25, respectively.

$$\text{minimize}\left(\text{CV} - \sum_{i \in I^-} \pi_i Y_i - \sum_{r \in R^{\text{T}}} \sum_{t \in T} b_{tr}^{\text{end}} \sum_{i \in I^-} \pi_{ri}^t x_t Y_i\right) \tag{6e}$$

subject to

$$\sum_{t \in T} x_t = 1 \tag{17a}$$

$$x_t = 1 \qquad t = \text{selected terminal} \tag{17b}$$

$$D_i \geq \sum_{t \in T} x_t d_{ti} \qquad \forall\ i \in I^- \tag{18}$$

$$\{D_j \geq D_i + d_{ij} - M_D(1 - S_{ij}) - M_D(2 - Y_i - Y_j)\} \quad \forall\ (i, j) \in I^- : i < j \tag{19a}$$

$$\{D_i \geq D_j + d_{ij} - M_D S_{ij} - M_D(2 - Y_i - Y_j)\} \quad \forall\ (i, j) \in I^- : i < j \tag{19b}$$

$$\text{CV} \geq \text{cf}_v + D_i + \sum_{t \in T} x_t d_{ti} - M_C(1 - Y_i) \qquad \forall\ i \in I^- \tag{20}$$

$$T_i \geq t^{\text{start}(-)} + \sum_{i \in I^-} Y_i \text{st}_i + \sum_{t \in T} x_t t_{ti} \qquad \forall\ i \in I^- \tag{21}$$

$$\{T_j \geq T_i + st_i + t_{ij} - M_T(1 - S_{ij}) - M_T(2 - Y_i - Y_j)\} \quad \forall\ (i, j) \in I^- : i < j \tag{22a}$$

$$\left\{T_i \geq T_j + \mathrm{st}_j + t_{ji} - M_T S_{ij} - M_T(2 - Y_i - Y_j)\right\} \quad \forall\ (i, j) \in I^-: i < j \tag{22b}$$

$$\mathrm{TV} \geq T_i + \sum_{t \in T} x_t t_{ti} - M_T(1 - Y_i) \qquad \forall\ i \in I^- \tag{23}$$

$$t_i^{\min} \leq T_i \leq t_i^{\max} \qquad \forall\ i \in I^- \tag{24}$$

$$\mathrm{TV} \leq t^{\max(-)} \tag{25}$$

$$\sum_{i \in I^+} Y_i l_i \leq q \tag{26}$$

4.2.3. The Slave Pick-Up and Delivery Problem. The objective of the slave problem aimed at designing a mixed pick-up and delivery route is stated by eq 6f and is subject to constraints (27)−(37). The formulation is similar to previous slave formulations but the routes can visit both pick-up and delivery locations. The objective function is the pricing reformulation of eq 6c and does not include the term related to inbound/outbound load-flows on the terminal, because the vehicle that picks up a load must deliver it and no cargo must be left on the vehicle at the end of the tour. According to eq 35, all mixed tours must be fulfilled within the $[0,t^{\max(-)}]$ timespan. Constraint (37a) forces both locations to be serviced by the same vehicle and, since a request load must be picked up before the unload activity, constraint (37b) sets the precedence relationship between both request vertexes.

$$\text{minimize}\left(\mathrm{CV} - \sum_{i \in I^+} \pi_i Y_i - \sum_{i \in I^-} \pi_i Y_i\right) \tag{6f}$$

subject to

$$\sum_{t \in T} x_t = 1 \tag{27a}$$

$$x_t = 1 \qquad t = \text{selected terminal} \tag{27b}$$

$$D_i \geq \sum_{t \in T} x_t d_{ti} \qquad \forall\ i \in I \tag{28}$$

$$\left\{D_j \geq D_i + d_{ij} - M_D(1 - S_{ij}) - M_D(2 - Y_i - Y_j)\right\} \quad \forall\ (i, j) \in I: i < j \tag{29a}$$

$$\left\{D_i \geq D_j + d_{ij} - M_D S_{ij} - M_D(2 - Y_i - Y_j)\right\} \quad \forall\ (i, j) \in I: i < j \tag{29b}$$

$$\mathrm{CV} \geq \mathrm{cf}_v + D_i + \sum_{t \in T} x_t d_{ti} - M_C(1 - Y_i) \qquad \forall\ i \in I \tag{30}$$

$$T_i \geq \sum_{t \in T} x_t t_{ti} \qquad \forall\ i \in I \tag{31}$$

$$\left\{T_j \geq T_i + \mathrm{st}_i + t_{ij} - M_T(1 - S_{ij}) - M_T(2 - Y_i - Y_j)\right\} \quad \forall\ (i, j) \in I: i < j \tag{32a}$$

$$\left\{T_i \geq T_j + st_j + t_{ji} - M_T S_{ij} - M_T(2 - Y_i - Y_j)\right\} \quad \forall\ (i, j) \in I: i < j \tag{32b}$$

$$\mathrm{TV} \geq T_i + \sum_{t \in T} x_t t_{ti} - M_T(1 - Y_i) \qquad \forall\ i \in I \tag{33}$$

$$t_i^{\min} \leq T_i \leq t_i^{\max} \qquad \forall\ i \in I \tag{34}$$

$$\mathrm{TV} \leq t^{\max(-)} \tag{35}$$

$$\sum_{i \in I^+} Y_i l_i \leq q \tag{36}$$

$$\left\{Y_i = Y_j\right\} \qquad \forall\ \tau = \{i, j\} \in \Gamma: i \in I^+ \cap \tau, j \in I^- \cap \tau \tag{37a}$$

$$\left\{S_{ij} = 1\right\} \qquad \forall\ \tau = \{i, j\} \in \Gamma: i \in I^+ \cap \tau, j \in I^- \cap \tau \tag{37b}$$

4.3. The Branching Strategy. The linear relaxation of the RMP is not necessarily an integer, and applying a standard branch-and-bound procedure to this problem with a given pool of columns may not guarantee an optimal solution.[17] Also, a column pricing favorably may exist but it may not be present in the RMP. Consequently, to determine the optimal solution, columns must be generated after branching. Ryan and Foster[29] proposed a branching strategy that is quite popular in column generation applications and fits in a natural way with the above formulations of the slave problems. The rule amounts to selecting two locations i and j and generating two branch-and-bound nodes: one in which locations i and j are serviced by the same vehicle, and the other where locations i and j are serviced by different vehicles. To enforce the branching constraints, rather than adding explicitly them to the master problem, the infeasible columns are eliminated from the columns-set considered in the branch-and-price node. Desrochers et al.[30] proposed another simple but effective branching strategy that consists of branching in the number of tours. If the master problem returns a solution that is fractional in the number of used tours k, it is natural to introduce a bound in the number of columns k. We branch on this number by creating two child nodes equivalent to the current subspace but with the addition of $\sum_r X_r \geq \mathrm{ceil}(k)$ and $\sum_r X_r \leq \mathrm{floor}(k)$ constraints to the respective master problems. This branching strategy should be effective when solving problems that include fixed costs in the column costs, because the total cost should be sensitive to the saving of a tour. We integrated both branching rules in a hierarchical way. The implemented branching scheme uses branching on the number of vehicles first and whenever this number has been fixed, we start to branch according the Ryan and Foster rule. The best first search was the node selection strategy.

4.4. Implementation Issues. The branch-and-price algorithm has been coded in GAMS 23.6.2 and integrates a CG routine into a branch-and-bound routine. Both GAMS routines were separately developed by Kalvelagen[31,32] and were integrated in this work to lead to the truncated branch-and-price procedure detailed above. Minor branching and assembling modifications were also introduced. The algorithm uses the CPLEX 11 as the MILP subalgorithm for generating columns and for computing upper and lower bounds. The algorithm run in a 2-core, 2.5 GHz, 6 MB RAM notebook and the mechanism settings used to solve the problems are summarized in Table 1.

Branch-and-price is an enumeration algorithm enhanced by fathoming based on bound comparisons. It is best to work with

Table 1. Settings Options of the Branch-and-Price Algorithm

option	value/comment
MILP solver	CPLEX 11
branching rule	on the number of tours + Ryan and Foster
nodes selection strategy	best first search
maximum CPU time per master−slave iteration (s)	30
multiple columns generated per iteration	yes
time-windows reduction	yes
maximum number of master−slave iterations per branch and price node	100 (root)/5 (no-root)
maximum number of branch-and-price inspected nodes	100
master problem	constraints (2) and (3): partitioning
columns pool	up to 10 000

the strongest bounds, although the mechanism can work with any bound. This leads to a tradeoff between the CPU time used in computing strong bounds and the size of the tree. To reduce the "tailing-off" effect, which consists of a very low convergence rate at the last iterations of the master−slave recursion, we ended it after 5 iterations in no-root nodes and used the bounds computed in such a way. Time-windows reduction and preprocessing were also fully used in order to increase the resolution efficiency. (See ref 27 for details on preprocessing and time-windows reduction rules tailored to slave problems similar to the ones defined in section 4.2.) Since the maximum number of nodes to inspect is bounded and the master−slave recursion is terminated after 5 iterations in no-root nodes, this incomplete procedure is of a heuristic nature. To provide an initial solution, feasible routes t−i−t, starting from any terminal $t \in T$, are generated and associated with each site $i \in \{I^+ \cup I^-\}$. From this initial routes-package, plus the set of all transfers, the linear RMP can compute the bounds to start the master−slave recursion.

5. COMPUTATIONAL RESULTS

The solution procedure was first tested with several forms of academic-type instances in order to evaluate its performance and to estimate an approximated measure of the quality of the provided solutions. Later, it was applied to a real case study that motivated the development of the model presented in section 3, as well as its associated solution procedure developed in section 4.

5.1. Testing Examples. To validate the entire procedure, we started our campaign of numerical experiments by solving some small-scale academic instances. These instances were generated by modifying some VRPTW instances that were proposed by Solomon.[33] Each Solomon problem has 100 customers, whose locations are generated in the Euclidean plane and are defined by the (X,Y) coordinates. The travel-time between locations is equal to the Euclidean distance. Problems with 25 or 50 customers may be generated by selecting the first 25/50 locations of the vertexes list and all instances are symmetrical. To adapt these examples to the above-described problem, several terminals were introduced in addition to the original depot on all Solomon R1 class problems. This group was selected because their time-windows lead to solutions involving a wide span of solution-shapes. The terminal locations in the same Euclidean plane of the Solomon examples are presented in Table 2. The testing instances were generated

Table 2. Euclidean Coordinates of the Terminals

terminal	X_{coord}	Y_{coord}	comments
$T1$	35	35	original depot
$T2$	15	20	new terminal
$T3$	15	50	new terminal
$T4$	50	50	new terminal
$T5$	50	20	new terminal

by using examples with 50 locations. The first 25 locations were considered as pick-up nodes, while the last 25 were considered as delivery locations. In this way, the first request is defined by the first pick-up location coupled to the first delivery site. The cargo assigned to the former site must be transferred to the later one.

In the examples considered, the service times st_i at each pickup/delivery location $i \in \{I^+ \cup I^-\}$ involve two components: a fixed preparation time, t_f = 10 time-units and a variable component that depends on the size of the load to deliver, t_v = 1 time-unit/time load; so $st_i = t_f + t_v |l_i|$. The terminals host a number of local vehicles with a capacity of q = 50 load-units and a fixed utilization cost of $cf_{v\text{-local}}$ = 20 Euclidean-units. All pick-up time windows were taken from the Solomon VRPTW instances. To ensure the feasibility of the delivery, the time windows of the source and destiny locations must be "compatibilized". In this work, the original time-windows of delivery sites were modified according the following rules:

$$a_j = a_i + st_i + \min_{t \in T}(t_{it} + qt_v + t_{transf} + qt_v + t_{tj}) \tag{38a}$$

$$b_j = b_i + st_i + \max_{t \in T}(t_{it} + qt_v + t_{transf} + \max_{t':t \neq t'} + qt_v + t_{tj})$$
$$\forall\ \tau = \{i, j\} \in \Gamma\colon i \in I^+ \cap \tau, j \in I^- \cap \tau \tag{38b}$$

These rules push forward the original delivery-time windows to allow the unload/load operations in the terminals and the long-haul trips performed between the pick-up and delivery phases of the problem. Other rules can be used and they mostly are dependent on the desired service level to customers. The time-span devoted to load/unload transshipment and long-haul routes is $t_{transf} = qt_v$ time-units and the trips between terminals are performed by vehicles with a fixed utilization cost $cf_{v\text{-long-haul}}$ = 40 Euclidean-units representing both transshipment costs and vehicle fixed costs. In addition, the Euclidean travel costs between terminals are increased by a coefficient c = 1.5. As long-haul routes are performed by faster vehicles, the travel time between terminals is considered as numerically equal to half the Euclidean distance between them. These instances were solved first with the option of mixed tours allowed (Mode 1) and later without such a possibility (Mode 2). In addition, to evaluate the consequences of changes on the terminals-network, such as the opening or closure of some terminals, we solved this 25-request problem-series using different number of terminals. This allows to estimate the savings provided by the option of using mixed tours on several terminals-networks configurations. The two-level branching strategy and both simple branching rules were tested, and the best results that were found are summarized in Table S1 in the Supporting Information. As it can be appreciated, the lower bound provided for an instance with the mixed trips option disabled is usually higher than the upper bound for the corresponding instance with mixed trips allowed. Thus, a sizable savings may be obtained while considering direct deliveries via a mixed route, because this option allows one to

avoid some cross-docking operations at terminals whenever some requests can be fulfilled via a mixed route. In addition, the number of mixed tours seems to be dependent on the time windows. Despite the complex multilevel networks involved in such types of problems, the gap remains below the 20% threshold in all Mode 1 instances. If necessary, the gap may be reduced at a higher computational cost by enlarging the size of the truncated search tree. Note that, despite the moderate number of transportation requests, these instances are quite complex, because they do not consider any delimitation between services areas. The delimitation of service areas naturally arise in practical scenarios but was not already considered.

To test the algorithm in larger and realistic instances, we consider now several scenarios for the delivery of 50 transportation requests by a carrier-system comprised of the 5 terminals detailed in Table 2. The cargo capacity for the local vehicles is increased up to $q = 75$ load-units in these examples. The instances are generated by considering the first 50 locations of the Solomon R1 problems as pick-up nodes and the last 50 sites as delivery locations. The first request is defined by the first pick-up site coupled to the first delivery location, and so on. For example, the load $l_1 = 10$ units taken from site n_1 ($Xn_1 = 41$; $Yn_1 = 49$) must be transferred to site n_{51} ($Xn_{51} = 49$; $Yn_{51} = 58$). The first scenario considers no overlapped service areas, while the second scenario allows a small area of overlapping. The third one increases the level of overlapping. The information about terminals service-areas is reported in Table 3, and the results for all solved examples are summarized in Table S2 in the Supporting Information.

Table 3. Service Areas for the Terminals in the Three Scenarios for the Examples with 50 Transportation Requests

terminal	function	service-area coordinates
		Scenario A
T1	hub	$[15 \leq X \leq 55; 15 \leq Y \leq 65]$
T2	hub	$[0 \leq X \leq 35; 0 \leq Y \leq 45]^a$
T3	hub	$[0 \leq X \leq 35; 35 \leq Y \leq 80]^a$
T4	hub	$[35 \leq X \leq 80; 35 \leq Y \leq 80]^a$
T5	hub	$[35 \leq X \leq 80; 0 \leq Y \leq 35]^a$
		Scenario B
T1	hub	$[15 \leq X \leq 55; 15 \leq Y \leq 65]$
T2	hub	$[0 \leq X \leq 35; 0 \leq Y \leq 35]$
T3	hub	$[0 \leq X \leq 35; 35 \leq Y \leq 80]$
T4	hub	$[35 \leq X \leq 80; 35 \leq Y \leq 80]$
T5	hub	$[35 \leq X \leq 80; 0 \leq Y \leq 35]$
		Scenario C
T1	hub	$[15 \leq X \leq 55; 15 \leq Y \leq 65]$
T2	hub	$[0 \leq X \leq 35; 0 \leq Y \leq 45]$
T3	hub	$[0 \leq X \leq 35; 35 \leq Y \leq 80]$
T4	hub	$[35 \leq X \leq 80; 35 \leq Y \leq 80]$
T5	hub	$[35 \leq X \leq 80; 0 \leq Y \leq 45]$

[a]Except the area overlapped with the service area of terminal T1.

It is clear from the comparison of solutions to the same instance but with different degrees of service-areas overlapping that the overlapping allows some routes to be saved, as a way to reduce the total system cost. So, when delimiting service areas, it is advisible to allow certain overlapping and give to the solution procedure the freedom to choose the location-to-vehicle-to-terminal allocation relationships. In some real situations, the delimitation naturally arises because the service areas are located far away. To illustrate the topology of the solution to a given instance, we detailed the solution provided by the algorithm to the instance R-112 (scenario C) in Table S3 in the Supporting Information and depicted it in Figure 3. The solution involves 11 pick-up routes, 12 transfers (4 cross-docking operations on a terminal + 8 long-haul routes), 11 delivery routes and a single pick-up and delivery route. Note in the figure that the delimitation of working areas is quite clear in the delivery stage but is less obvious than in the pick-up phase. In this phase, the overlapping of areas allows one to save some local tours that, in the case of rigid areas delimitation, must be used. In the end, whenever customers' time windows are quite constraining, it should be convenient to enlarge the overlapping between service areas. The detail of costs for the solution is reported in Table S4 in the Supporting Information.

Table 4 details the roadmaps and schedules provided to the truck drivers for a pick-up route, for a mixed pick-up and delivery route, and for a delivery route. Any driver must be provided a similar roadmap, regardless of the route type.

5.2. A Case Study with Real Data.

After the extensive testing of the previous section, we illustrate the use of the solution procedure on a case study with real data. A transportation company from Santa Fe (Argentina) that provides distribution services of nonperishable products to several industrial (woodworking companies, food companies, small-scale industries) and service companies (supermarkets, retailers) in the urban Santa Fe area and surroundings gave us some of its daily operational data.

The daily operation here considered involves the use of several vans based on two depots (Central depot and S. Tomé depot) that exchange cargo by using a single truck once a day. The truck, which is based in the Central depot, go to the secondary one and returns to the Central depot again. Vans are used to collect/deliver small cargo and their maximum volumetric capacity is $q = 7.5$ m^3. The truck capacity is large enough to be considered unconstraining. Service times at pick-up/delivery stops are considered approximately constant, $st_i = 20$ min; the average urban-travel speed is quite difficult to estimate, but it is conservatively assumed to be 20 km/h. The case study uses data from a typical working day and involves the fulfillment of 44 transportation requests within the day. Usually, the company performs pick-up activities during "the morning" and delivery during "the afternoon" to allow some consolidation work between both stages and to avoid cargo warehousing on depots at night. Sometimes, the pick-up tasks may be performed during afternoon and the delivery during the morning of the next day but we used data for the first modality, because, in that case, some vans are allowed to perform both pick-up and delivery tasks. Time windows usually are not considered and sometimes they can be assigned just to a few "important" clients and only for pick-up activities. We estimated the distance (in kilometers) between client locations and between these locations and both depots by using the Manhattan distance formula jointly with the client locations on the city map. The datasheet for the case is presented in the Supporting Information (Table S4). A fixed van utilization cost of $cf_v = \$200$ and a unit-distance cost of \$10/km are considered here. The round trip, "Central depot−S. Tomé−Central depot", which includes transportation and workload costs on both depots, have an associated cost of $cf_{\text{long-haul}} = \$3400$/day. Cargo transhipment costs on each depot are $cf = \$400$/day.

We applied the solution algorithm developed above to that case study and generated the solution to be detailed next. The

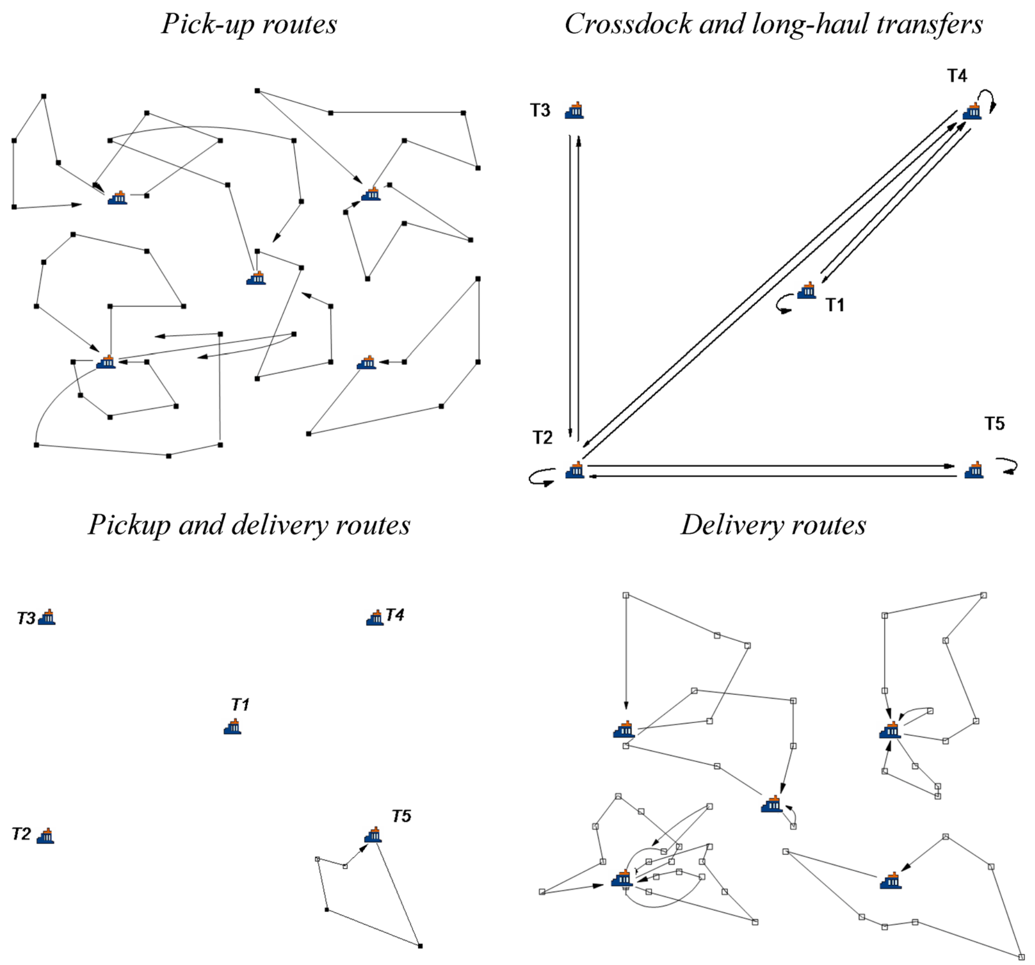

Figure 3. Routes specified by the solution to example R-112 in scenario C.

solution was obtained within 2889 s (gap = 9.8%) and involves 7 pick-up tours, 6 delivery tours, 4 mixed tours, and 2 transfer-trips. It implies a total cost of $17630, and it is summarized in Tables 5−8 and illustrated in Figure 4.

The computed plan can be described as follows. During the morning, five vans depart from the Central depot and return there to unload the collected cargo (26.6 m^3). Two vans based at the secondary depot (S. Tomé) perform similar tasks and consolidate their 14.2 m^3 of cargo. Then, a large truck moves 7.5 m^3 of the load from the Central depot to the secondary depot and return to the Central depot with 14.2 m^3 of cargo consolidated in the secondary one (i.e., the same truck performs both long-haul trips).

The afternoon-stage is devoted to the distribution of cargo, mainly from the central depot (33.3 m^3). Just one delivery tour based in S. Tomé is performed to deliver 7.5 m^3 of cargo. Meanwhile, four vans were allowed to deliver the collected cargo without returning to the base depots. It can be observed that no cross-docking operations were performed in the secondary depot. There, the load coming from the main depot was received and delivered to the final destinations.

6. CONCLUSIONS

We have developed a truncated branch-and-price solution algorithm to efficiently design a transportation agenda for a less-than-truckload (LTL)-type problem involving the fulfillment of a list of transportation requests by choosing between different delivery options. The problem arises from a third-party logistics (3PL) company that provides small-size cargo transportation services to production and services companies. Several shipping alternatives were considered in the problem: a direct delivery to the destination using a single vehicle; a delivery via transshipment on a terminal; and a three-stage delivery option, which includes a pick-up step, a long-haul route between two terminals, and, finally, the delivery. The problem was first modeled as a set partitioning problem with an additional set of transfer constraints. The model was later reformulated and embedded into a column generation (CG) procedure to develop the branch-and-price solution-mechanism for the problem. The proposed mechanism has the following original features:

(i) It reorder the long-haul transfer constraints to express them as covering constraints to add to the partitioning constraints for pick-up and delivery locations.

(ii) It utilizes multiple route−generator problems at the slave level of the CG procedure. Since the problem involves several types of routes, specific integer-linear programs for each route type were also developed.

(iii) The pricing problems were formulated as integer-linear programs and solved by a branch-and-cut solver, in an attempt to maximize the solutions diversification in order to obtain a maximum number of elementary columns per master−slave iteration.

Some standard options were also taken: branching on the number of tours was selected as a higher-level branching-rule to

Table 4. Road Maps for (a) a Local Pick-up Route Departing (Returning) from (to) Terminal T4, (b) a Local Delivery Route Departing (Returning) from (to) Terminal T4, and (c) a Mixed Pick-Up and Delivery Route Departing (Returning) from (to) Terminal T5

location	arrival time	departure time	activity	onboard load
Local Pick-Up Route Departing (Returning) from (to) Terminal T4				
T4		0.0	departure	0
r9	47.0	73.0	pickup	16
r34	84.2	108.2	pickup	30
r35	118.4	136.4	pickup	38
r20	154.4	173.4	pickup	47
r32	184.1	217.1	pickup	70
T4	241.4	321.4	total unload	0
(b) Local Delivery Route Departing (Returning) from (to) Terminal T4				
T4		359.0	departure	73
r29	366.3	385.3	delivery	64
r28	391.0	417.0	delivery	48
r21	433.5	454.5	delivery	37
r15	468.8	482.8	delivery	29
r16	496.4	525.4	delivery	10
r1	540.4	560.4	delivery	0
T4	568.4			
(c) Mixed Pick-Up and Delivery Route Departing (Returning) from (to) Terminal T5				
T5		0.0	departure	0
r23	36.0	75.0	pickup	29
r22	153.0	181.0	pickup	47
r23	325.5	364.5	delivery	18
r22	509.0	537.0	delivery	0
T5	542.0			

Table 5. Pick-Up Tours for the Case Study

	Central Depot					S. Tomé	
	Tour 1	Tour 2	Tour 3	Tour 4	Tour 5	Tour 1	Tour 2
	n10	n31	n7	n30	n45	n16	n44
		n40	n8	n20	n19	n21	n38
		n25	n48	n50	n18	n39	n42
		n24	n46	n13			n15
		n33	n36	n17			n14
cost ($)	274	1024	877	920	664	901	431
tour time (min)	147	265	279	258	153	200	146
unload time (min)	167	285	299	278	173	220	166
cargo (m^3)	1.5	6.0	6.7	7.2	5.2	6.7	7.5

Table 6. Requests Moved on Depots and between Both Depots for the Case Study

crossdocking operations	requests moved	cargo moved (m^3)	cost ($)
Central depot	n10−n31−n40−n33−n8−n46−n30−n20−n13−n45−n19−n18	19.1	400
long-haul routes	**requests transferred**	**cargo moved (m^3)**	**cost ($)**
Central−S. Tomé	n25−n24−n7−n48−n36−n50−n17	7.5	1700
S. Tomé−Central	n16−n21−n39−n44−n38−n42−n15−n14	14.2	1700

Table 7. Delivery Tours for the Case Study

	Central Depot					S. Tomé
	Tour 1	Tour 2	Tour 3	Tour 4	Tour 5	Tour 1
	n30	n38	n33	n31	n10	n57
	n18	n46	n14	n21	n44	n24
	n19	n42	n13	n15	n39	n25
	n40	n45		n16		n17
		n58				n48
		n20				n50
						n36
cost ($)	834	743	456	787	562	877
start time (min)	481	445	558	522	506	422
end time (min)	690	674	669	720	666	705
cargo (m^3)	6.7	6.7	5.7	7.2	7.0	7.5

Table 8. Mixed Pick-Up and Delivery Tours for the Case Study[a]

	Central Depot		S. Tomé	
	Tour 1	Tour 2	Tour 3	Tour 1
	$n27^+$	$n11^+$	$n32^+$	$n23^+$
	$n28^+$	$n49^+$	$n9^+$	$n22^+$
	$n26^+$	$n47^+$	$n35^+$	$n43^+$
	$n12^+$	$n49^-$	$n34^+$	$n41^+$
	$n26^-$	$n11^-$	$n29^+$	$n37^+$
	$n27^-$	$n47^-$	$n29^-$	$n43^-$
	$n28^-$		$n99^-$	$n41^-$
	$n32^-$		$n35^-$	$n37^-$
			$n34^-$	$n23^-$
			$n32^-$	$n22^-$
cost ($)	936	1034	1227	1084
end time (min)	666	611	677	661
cargo (m^3)	7.0	7.2	7.3	6.8

[a]Superscripted plus signs denote pick-up stops; superscripted minus signs denote delivery stops.

explore a finite branch-and-price tree. After fixing the number of vehicles, the algorithm starts to branch according the Ryan and Foster rule. Both rules can also operate as a lone branching rule.

The proposed algorithm was validated by first solving some moderately sized academic-type instances involving 25 transportation requests and networks with up to 4 terminals over a nonpartitioned service area. The problems were solved with and without the possibility of using mixed pick-up and delivery routes, in order to compare results and estimate the savings provided by such a routing option. The comparison showed that sizable cost savings may be provided by the option of using mixed tours in numerous instances. Later, the procedure was tested on larger scenarios with 50 requests and up to 5 terminals. Three scenarios were defined to solve such examples. In the first one, no overlapping between service areas of terminals were considered. The second scenario allows a certain level of overlapping, and the third one increases that level of overlapping. The quality of solutions, as well as the consumed CPU times, were good in all cases. The worst-case duality gap was 4.28% (average gap = 2.24%) in the examples of the first scenario; 10.39% (average gap = 6.73%) for the examples of the second scenario; and 14.03% (average gap = 9.14%) for the last scenario. Remarkably, all examples were solved within less than 1 h and most of them within less than half an hour.

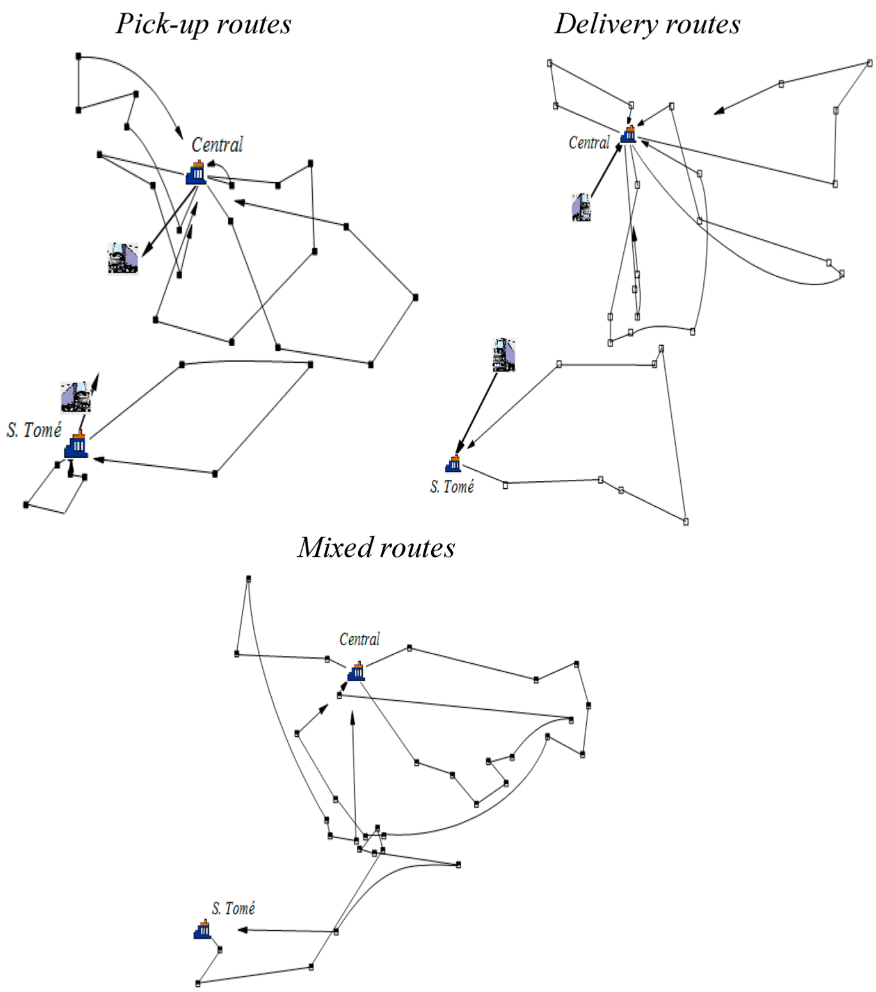

Figure 4. Routes specified by the solution to the case study.

Finally, we presented the realistic case study that motivated the development of the model and of the solution procedure developed above. In the real-world case, the delimitation of the service areas is quite obvious, which allows one to massively reduce the solution space. The solution to the case study was also detailed.

Some issues not considered in this work may deserve future research:

(i) To enlarge the size of solved examples and/or reduce the duality gap, a more elaborate and efficient method to provide elementary routes to the RMP may be developed.

(ii) Long-haul trips should also consider capacity constraints for the large carriers.

(iii) Timing constraints may be introduced in the set partitioning problem to avoid the rigid delimitation between pick-up, delivery, and transfer phases.

(iv) A more general problem considering pure delivery and pure pick-up requests may be considered. This is done with the purpose of considering cases with pick-up tasks performed during the day/afternoon and the delivery during the morning of the next day.

(v) The extension to more-generalized supply-chain problems, where the concept of transportation requests is replaced by sets of cargo-source and cargo-sink locations, should be also considered.

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

This material is available free of charge via the Internet at http://pubs.acs.org/.

■ AUTHOR INFORMATION

Corresponding Author

*Tel.: +54 342 4559175. E-mail: rdondo@santafe-conicet.gov.ar.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by CONICET (Grant No. PIP 2221) and Secretaría de Estado de Ciencia y Producción de la Provincia de Santa Fe (Grant No. 2010-161-12). The authors want to thank Transporte Poccia Hermanos for supporting our grant application and for providing the daily operational data for the case study.

■ NOMENCLATURE

Sets

A = arcs of the routes network
I^+ = pick-up sites
I^- = delivery sites
R^T = transfers

R^+ = pick-up routes
R^- = delivery routes
R^{+-} = mixed pick-up and delivery routes
T = terminals
Γ = transportation requests

Parameters

a_{ir} = binary parameter denoting that site i is visited by route r
b_{tr} = binary parameter denoting that route r starts/ends in terminal t
b_{tr}^{start} = binary parameter denoting that long-haul route r starts on terminal t
b_{tr}^{end} = binary parameter denoting that long-haul route r ends on terminal t
cf_v = fixed cost of using a local vehicle
c_r = cost of route r
d_{ij} = distance between locations i and j
l_i, l_j, l_{ij} = load to pick-up from site i and to deliver to site j
M_C = upper bound for the travel cost (C)
M_D = upper bound for the traveled distance (D)
M_T = upper bound for the travel time (T)
q = transport capacity of the local vehicles
st_i = stop time at the pick-up/delivery site i
$t^{\max(+)}$ = maximum allowed routing time for the pick-up phase
$t^{\text{start}(-)}$ = start time of the delivery phase
$t^{\max(-)}$ = maximum allowed routing time for the delivery phase
$t_i^{\min}$ = earliest arrival time at the pick-up/delivery site i
$t_i^{\max}$ = latest arrival time at the pick-up/delivery site i
x_t = binary parameter stating that terminal t belongs to the route designed by a slave routes-generator problem
α_{ir}^{+} = binary parameter related to the unload on terminal t of the cargo picked up on site i
α_{ir}^{-} = binary parameter related to the load on terminal t of the cargo to deliver to site i
π_i^+ = price associated with pick-up site i
π_i^- = price associated with delivery site i
π_t = price associated with terminal t

Binary Variables

X_r = variable denoting that route r belongs to the optimal subset of feasible routes
S_{ij} = variable for sequencing locations i and j
Y_i = variable used to determine that site i belongs to the route designed by a slave route−generator problem

Continuous Variables

CV = total cost of the route designed by a slave route−generator problem
D_i = distance traveled to reach the pick-up/delivery site i
T_i = time spent to reach the pick-up/delivery site i
TV = time spent on the route designed by the slave route−generator problem

■ REFERENCES

(1) Dondo, R.; Méndez, C.; Cerdá, J. The Multi-Echelon Vehicle Routing Problem with Cross-Docking in Supply Chain Management. *Comput. Chem. Eng.* **2011**, *35* (12), 3002−3024.

(2) Dondo, R.; Méndez, C.; Cerdá, J. Managing Distribution in Supply Chain Networks. *Ind. Eng. Chem. Res.* **2009**, *48* (22), 9961−9978.

(3) Grossmann, I. Challenges in the New Millennium: Product Discovery and Design, Enterprise and Supply Chain Optimization, Global Life Cycle Assessment. *Comput. Chem. Eng.* **2004**, *29* (1), 29−39.

(4) Erera, A.; Hewitt, M.; Savelsbergh, M.; Zhang, Y. Creating schedules and computing operating costs for LTL load plans. *Comput. Oper. Res.* **2013**, *40*, 691−702.

(5) Amaro, A.; Barbosa-Povoa, A. Optimal supply chain management with detailed scheduling. *Ind. Eng. Chem. Res.* **2008**, *47* (1), 116−132.

(6) Powell, W. A local improvement heuristic for design of less-than-truckload motor carrier networks. *Transport. Sci.* **1986**, *20* (4), 246−257.

(7) Powell, W. B.; Sheffi, Y. The load planning problem of motor carriers: Problem description and a proposed solution approach. *Transport. Res., Part A* **1983**, *17* (6), 471−480.

(8) Powell, W. B.; Sheffi, Y. Design and implementation of an interactive optimization system for network design in the motor carrier industry. *Oper. Res.* **1989**, *37* (1), 12−29.

(9) Powell, W.; Koskosidis, I. Shipment routing algorithms with tree constraints. *Transport. Sci.* **1992**, *26*, 230−45.

(10) Farvolden, J.; Powell, W. Subgradient methods for the service network design problem. *Transport. Sci.* **1994**, *28*, 256−72.

(11) Kuby, M.; Gray, R. The Hub Network Design Problem with Stopovers and Feeders: The case of Federal Express. *Transport. Res., Part A* **1993**, *27* (1), 1−12.

(12) Lin, C.; Chen, S. The Hierarchical Network Design Problem for Time-Definite Express Common Carriers. *Transport. Res., Part B* **2004**, *38* (3), 271−283.

(13) Estrada, M.; Robusté, F. Long-Haul Shipment Optimization for Less-Than-Truckload Carriers. *Transport. Res. Rec.: J. Transport. Res. Board* **2009**, *2091*, 12−20.

(14) Barcos, L.; Rodríguez, V.; Álvarez, M. J.; Robusté, F. Routing design for less-than-truckload motor carriers using Ant Colony Optimization. *Transport. Res., Part E* **2010**, *46*, 367−383.

(15) Jarrah, A.; Johnson, E.; Neubert, L. Large-scale, less-than-truckload service network design. *Oper. Res.* **2009**, *57* (3), 609−625.

(16) Toptal, A.; Bingöl, S. Transportation pricing of a truckload carrier. *Eur. J. Oper. Res.* **2011**, *214*, 559−567.

(17) Barnhart, C.; Johnson, E.; Nemhauser, G.; Savelsbergh, M.; Vance, P. Branch and Price. Column Generation for Solving Huge Integer Programs. *Oper. Res.* **2000**, *48* (3), 316−329.

(18) Vanderbeck, F. On Dantzig−Wolfe-Decomposition in Integer Programming and Ways to Perform Branching in a Branch-and-Price Algorithm. *Oper. Res.* **2000**, *48*, 111−128.

(19) Desrosiers, J.; Lubbecke, M. Branch-Price-and-Cut algorithms. In *Wiley Encyclopedia of Operations Research and Management Science*; Cochran, J., Ed.; John Wiley & Sons, Inc.: New York, 2010.

(20) Bettinelli, A.; Ceselli, A.; Righini, G. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. *Transport. Res., Part C* **2011**, *19* (5), 723−740.

(21) Santos, F.; Mateus, G.; da Cunha, A. A branch-and-price algorithm for a vehicle routing problem with cross-docking. In *LAGOS'11—VI: Latin-American Algorithms, Graphs and Optimization Symposium*; Electronic Notes in Discrete Mathematics, Vol. *37*; 2011; pp 249−254.

(22) Santos, F.; Mateus, G.; da Cunha, A. The Pickup and Delivery Problem with Cross-Docking. *Comput. Oper. Res.* **2013**, *40*, 1085−1093.

(23) Irnich, S. and Desaulniers, G. Shortest Path Problems with Resource Constraints. In *Column Generation*; Desaulniers, G., Desrosiers, J., Solomon, M., Eds.; Springer: New York, 2005; pp 33−65.

(24) Feillet, D.; Dejax, P.; Gendreau, M.; Gueguen, C. An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to Some Vehicle Routing Problems. *Networks* **2004**, *44* (3), 216−229.

(25) Chabrier, A. Vehicle routing problem with elementary shortest path based column generation. *Comput. Oper. Res.* **2006**, *33* (10), 2972−2990.

(26) Righini, G.; Salani, M. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimiz.* **2006**, *3*, 255−273.

(27) Dondo, R. A New MILP Formulation to the Shortest Path Problem with Time Windows and Capacity Constraints. *Latin Am. Appl. Res.* **2012**, *42*, 257−265.
(28) Gualandi, S.; Malucelli, F. Constraint Programming-based Column Generation. *Annals Oper. Res.* **2013**, *204*, 11−32.
(29) Ryan, D.; Foster, B. An integer programming approach to scheduling. In *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*; Wren, A., Ed.; North-Holland: Amsterdam, 1981; pp 269−280.
(30) Desrochers, M.; Desrosiers, J.; Solomon, M. A new optimization algorithm for the vehicle routing problem with time windows. *Oper. Res.* **1992**, *12* (2), 342−354.
(31) Kalvelagen, E. *Columns Generation with GAMS*. Downloaded from http://www.amsterdamoptimization.com/pdf/colgen.pdf, 2003.
(32) Kalvelagen, E. *Some MINLP Solution Algorithms*. Downloaded from http://www.amsterdamoptimization.com/pdf/minlp.pdf, 2003.
(33) Solomon, M. Algorithms for the vehicle routing and scheduling with time window constraints. *Oper. Res.* **1987**, *15*, 254−265.