Computers and Chemical Engineering 80 (2015) 15–29

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

A branch-and-price approach to evaluate the role of cross-docking operations in consolidated supply chains

Mariana Cóccola, Carlos A. Méndez, Rodolfo G. Dondo*

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

ARTICLE INFO

Article history:
Received 24 September 2014
Received in revised form 9 April 2015
Accepted 30 April 2015
Available online 9 May 2015

Keywords:
Supply chain
Cross-docking
Direct delivery
Column generation
Branch-and-price
Decomposition

ABSTRACT

Supply-chain management and optimization aims at reducing costs and inventories. One way to increase the supply-chain efficiency is to use cross-docking for consolidating shipments from different suppliers. Cross-docking is a warehousing strategy used in logistics that consists on moving goods from suppliers to customers through a cross-dock facility. The employment of this strategy must be carefully evaluated because sometimes transportation requests can be better directly moved from source-sites to destination. A realistic problem studying the convenience of direct delivery, avoiding some cross-docking transfers, is here discussed. An efficient methodology for finding (near)optimal solutions is also described. The methodology is based on the use of column generation embedded into an incomplete branch-and-price tree. The approach provides (near)optimal solutions by solving the column generation sub-problems without necessarily considering all unexplored nodes in the search-tree. Finally, we show computational results on numerous test problems and on four configurations of the addressed case study.

1. Introduction

Nowadays, supply chain management and optimization is a critical aspect of modern enterprises and a very active research area (Papageorgiou, 2009). The goal of the supply chain planning and plant scheduling problem consists on determining the optimal manufacturing and network distribution policies for the entire supply chain (SC) of a company in order to fulfill a pre-established economic objective. Indeed from a multisite perspective, this integration problem is even more challenging, since it requires integration across both spatial and temporal scales (Grossmann, 2012). Chemical and industrial companies usually carry out a series of activities such as purchasing raw materials from suppliers, manufacturing and storing end-products at intermediate facilities to later deliver them to final customers. Suppliers, manufacturers, warehouses and customers are the major components of an typical SC carrying goods from the upstream to the downstream side of the SC (Dondo et al., 2011). Supply chain management aims to control in the most efficient way the goods flow through the SC. An usual way to increase the efficiency of the SC is to outsource the movement of shipments on third parts logistics companies (3PL) that operate with a very high efficiency level. Small scale manufacturing companies usually lack resources to develop their own logistics leg and therefore are forced to outsource. In those cases, 3PL companies are required to consolidate shipments from different suppliers. After consolidating and sorting goods according to their destinations, incoming shipments move across the crossdock (CD) to exit doors where they are loaded onto outbound trucks that start their delivery routes. So, 3PL companies usually utilize cross-docking to consolidate shipments in order to efficiently operate the whole system. The logistic operator must coordinate truck loading and unloading operations with inbound and outbound vehicle routes. The more coordinated these issues are, the more cost and time effective the system should be. Nevertheless, it may not be convenient to force the transshipment of all cargo on the CD if the source and destiny locations of some requests are nearly placed. In that case, the orders can be driven directly from the pickup place to the delivery location rather than moving first to the CD platform.

The so called pick-up and delivery problem with cross-docking (PDPCD), first introduced by Santos et al. (2013) deal with the integration of vehicle routing and cross-docking operations, allowing some vehicles to avoid the stop at the CD if this is convenient to reduce transportation costs. The problem simultaneously considers the following logistics subproblems: (i) the pickup vehicle routing problem; (ii) the loads exchange between vehicles on the CD; (iii) the delivery vehicle routing problem; and (iv) the pickup and delivery problem for orders directly driven from the pick-up site to destiny. All subproblems should be coordinated in order to optimize the material flow from suppliers to destination locations. The coordination of pick-up and delivery routes jointly with the use

* Corresponding author. Tel.: +54 342 4559175.
E-mail address: rdondo@santafe-conicet.gov.ar (R.G. Dondo).

http://dx.doi.org/10.1016/j.compchemeng.2015.04.039

Nomenclature

Sets
A arcs of the routes network
I^+ pick-up sites
I^- delivery sites
R^{+-} pick-up and delivery routes
R^+ pick-up routes
R^- delivery routes
Γ transportation requests

Parameters
a_i earliest arrival time at the pick-up/delivery site i
a_{ir} binary parameter denoting that site i is visited by the route r
b_i latest arrival time at the pick-up/delivery site i
cf_v fixed cost of using a vehicle
c_r cost of the route r
$\hat{c}_r^{+-}$ reduced cost for a mixed route
$\hat{c}_r^+$ reduced cost for a pick-up route
$\hat{c}_r^-$ reduced cost for a delivery route
d_{ij} distance between the locations i and j
l_i, l_j, l_{ij} load to pick-up from the site i and to deliver to the site j
M_C, M_D, M_T upper bounds for the travel cost (C), the traveled distance (D) and the travel time (T)
q transport capacity of vehicles
st_i stop time at the pick-up/delivery site i
t^{max} maximum allowed routing time
$t_{ir}{}^-$ start time of cargo on the cross-dock for the load to deliver to site i
$t_{ir}{}^+$ unload time on the cross-dock for the cargo collected on site i
$\pi_i{}^+, \pi_i{}^-$ price associated to the pick-up/delivery site i
$\pi_\tau{}^t$ price associated to the loading/unloading time for request τ

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

Continuous variables
CV total cost of the route designed by a slave routes-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
$T_{end}{}^+$ end-time of unload activities on the CD
$T_{start}{}^-$ start-time of load activities on the CD
$T_i{}^+$ end-time for the unload of cargo from site i in the cross-dock
$T_i{}^-$ start-time for delivery of cargo to site i in the cross-dock
TV time spent by the route designed by a slave routes-generator problem

of mixed pick-up and delivery routes may produce a significant improvement on the overall distribution efficiency. In this work, we study, the coordination on the CD of unload and load tasks and the possibility of also using direct delivery trips from a 3PL point of view. The logistic operator is required to consolidate on the CD shipments from different sources to later deliver them to final destinations, taking also into account the possibility of direct delivery of some requests. The problem studied can be viewed as a variation of the PDPCD defined by Santos et al. (2013) that explicitly considers the time coordinating constraints between unload times from inbound vehicles and loading times on outbound vehicles. While the PDPCD considers a fixed fleet with a given number of vehicles, the problem here researched considers the fleet size as a variable derived from the problem solution. Also it considers delivery of stored loads from a source location and reverse logistics transport of some goods back to a depot. The problem is sketched in Fig. 1.

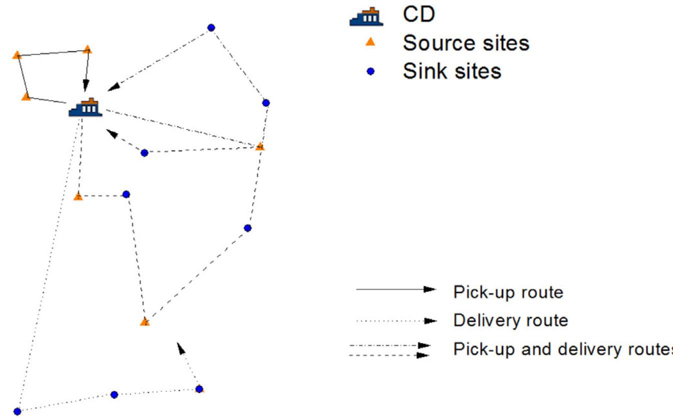


Fig. 1. Illustrating the problem.

The remain of the paper is organized as follows. In Section 2, we review the literature on issues related to the problem studied. The problem is described and formulated in Section 3. The incomplete branch-and-price methodology devised to solve the problem is detailed in Section 4. Numerical results on examples of the literature and on a case study are presented in Section 5 and the conclusions are outlined in Section 6.

2. Literature review

The need for good and, if possible, optimal solutions for routing problems has motivated, over the past decades, the development of an impressive number of solution algorithms, both exact and heuristics. Reviews on the subject could be found in Bodin et al. (1983), Ball et al. (1995), Desrosiers et al. (1995) and Ahuja et al. (2002). As the computing power increased and the solution techniques evolved, realistic and complex problems have been benefited from such developments. The integration of routing planning with production planning, to tackle sophisticated supply chains, was one of such developments. In this way, the integration of cross-docking with vehicles routing problems naturally arises. Cross-docking has already been applied in the1980s by Wal Mart but it has attracted attention from academia much later and mostly during the recent years (Van Belle et al., 2012). During the last years, a considerable number of papers on the subject have been published (Apte and Viswanathan, 2000) and because of the growing interest from industrial companies, more research on this topic should be expected. At the tactical and operational levels, contributions related to truck scheduling were reviewed by Boysen and Fliedner (2010). Van Belle et al. (2012) reviewed also numerous additional issues related to cross-docking as the physical and operational characteristics of the CD, the location and layouts of the CD, the associated vehicle routing problems and the door-to-vehicle assignment problems. Numerous mixed integer programming models for trucks scheduling problems of small or medium sizes, and meta-heuristic approaches for large-size case studies have been recently proposed (Tsui and Chang, 1992; Yu and Egbelu, 2008; Li et al., 2009; Boysen, 2010; Boloori Arabani et al., 2011) but fewer papers have dealt with the integration of vehicle routing with cross-docking. In these scenarios, freight destined to a CD are picked-up at various locations, and delivered to multiple locations after consolidation at the CD. Both the pickup and the delivery process can be seen as a vehicle routing problem and some studies consider cross-docking and vehicle routing simultaneously. To our knowledge, an early approach to this problem was developed by Lee et al. (2006). The aim of this problem is to find routes for pick-up and delivery activities minimizing the sum of transportation cost and fixed vehicles-costs. It was assumed that all pick-up vehicles should arrive at the cross-dock simultaneously to prevent waiting times for outbound trucks. While this hypothesis may be valid in some cases, it is not generally true. The authors presented an integer programming model which, however, seems unsatisfactory to solve the problem. So, a tabu-search algorithm was proposed to find solutions. Liao et al. (2010) proposed another tabu-search algorithm to solve the same problem and Wen et al. (2009) defined the so-called vehicle routing problem with cross-docking (VRPCD). In the problem, orders from suppliers must be picked-up by a homogeneous fleet of vehicles, consolidated at a cross-dock and delivered to customers by the same set of vehicles. During the consolidation, goods are unloaded from the inbound vehicles and reloaded onto outbound vehicles. The authors assume that the duration of the unloading tasks consists of a fixed time for preparation and a time-length proportional to the load size. It is also assumed that, if the delivery will be executed by the same vehicle used for pick-up, the unloading is not necessary. A time window is defined for all suppliers and customers. In cases without consolidation, the solution of this problem can be found by solving two vehicle routing problems but because of the consolidation, the pickup and delivery routes are not independent and must be coordinated. The authors presented a mixed integer programming formulation of the problem in which the objective is to minimize the total travel time. As the formulation is, in practice, insolvable because it contains many variables and constraints, the authors proposed to use tabu-search embedded within an adaptive memory procedure. This method was tested on instances involving up to 200 supplier–customer transportation requests. Dondo and Cerdá (2013) introduced a monolithic formulation for the VRPCD that determines pickup/delivery routes simultaneously with the truck scheduling at the CD. To accelerate the solution of the problem, constraints mimicking the widely known sweep algorithm were incorporated into the rigorous model. Instances with up to 50 transportation requests were solved in less than 3 h with this hybrid procedure. Santos et al. (2011) considered a slightly different VRPCD. Time windows were neglected and costs were added in the objective function to consider goods movements from a vehicle to another one at the CD. Later, Santos et al. (2013) introduced the so-called pick-up and delivery problem with cross-docking that in addition to pick-up and delivery routes considers the possibility of using mixed tours involving both pick-up and delivery activities. Since usual models that deal with the integration of vehicle routing and cross-docking impose that every vehicle must stop at the dock even if the vehicle collects and delivers the same requests, the authors allowed vehicles transporting such requests to avoid the stop at the CD to reduce transportation costs. The authors developed a branch-and-price procedure that was able to solve to (near)optimality instances with up to 50 requests.

3. Problem description, definition and formulation

A logistic operator usually provides convey services to production and services companies during a specified time period as, for example, on a daily basis. The objective is the transportation of goods from suppliers to end-locations through a cross-dock facility in order to satisfy a set of customer requests at minimum transportation cost. Each request includes the shipment size and location of the related pickup and destination sites. The modality includes many practical variations but in the case here studied, the system usually operates as follows: during a given time-horizon some vehicles depart from the cross-dock facility, service the assigned pickup nodes and return for unloading the collected goods on the cross-dock. After completing offload operations at the CD, a vehicle can immediately start reloading orders for moving them to their destinations. This CD operates as sorting and consolidation center and as a loading/unloading facility for inbound and outbound freight. Consolidation at the CD requires freight to be cross-docked. After consolidation and transshipments, the vehicles go from the CD to the assigned destinations. Since a shipment can be picked-up and delivered by the same carrier if the source and destination places are nearly located, some vehicles also can perform both pick-up and delivery activities along their designed tours. Vehicles may also deliver cargo already inventoried in the central depot. Service times at each pickup/delivery location have two components; a fixed time for preparation and a variable component that depends on the size of the load to be picked up or delivered. Similar stop-times are incurred in the CD. The solution to the problem must define a delivery-agenda stating the way freight is routed from origins to destinations for each request. The problem studied resembles the PDPCD defined by Santos et al. (2013) but differs from it in the following issues: (i) it explicitly considers the time coordinating constraints between unloading times from inbound vehicles and loading times on outbound vehicles i.e. the unloading end-time on the CD for picked goods and the loading start-time of goods to deliver are explicitly stated on constraints of the master problem of a column generation approach. (ii) It considers delivery of stored requests and the transport of some loads from several sites to the CD. Usually a transportation request defines a shipment size and the location of the related pickup and destination sites. This definition may include cases where the pickup site or the delivery location is the CD. This view, that can also be taken into account by the PDPCD, allows to solve scenarios considering the delivery of cargo already inventoried in the CD facility and/or the storing of loads on the CD. This view just needs to place an origin or destination node in the CD. (iii) Santos et al. (2013) considered a fixed fleet of $|V|$ vehicles and costs per cargo exchanges between vehicles. In our approach, the number of used vehicles is free and is derived from the problem solution. This view resembles the working mode of logistic operators that usually adjust the number of used trucks as a function of the volume of cargo to move. The problem studied is formally defined as follows:

Let $\boldsymbol{G}[I;\ A]$ a directed graph defined by the locations set $I=\{\mathrm{CD}\cup I^{+}\cup I^{-}\}$ and the network $A=\{a_{ij}\colon i, j\in I\ (i\neq j)\}$. The set I contains the cross-dock CD, the pick-up nodes $I^{+}=\{i_1, \ldots, i_n\}$ and the delivery nodes $I^{-}=\{i'_1, \ldots, i'_n\}$. A request $\tau=\{i, j\}$ of a request list $\Gamma=\{\tau_1, \ldots r_n\}$ consists of a demand of a transportation service from the origin-node(s) $i\in\{I_{\tau}{}^{+}\subset I^{+}\cup \mathrm{CD}\}$ to the destination node(s) $j\in\{I_{\tau}{}^{-}\subset I^{+}\cup \mathrm{CD}\}$ for a given load l_i. Each arc $a_{ij}\in A$ have associated an non-negative cost c_{ij} and an associated non-negative travel-time t_{ij}. The service time on each node $i\in I$ is computed as $(st_i+|l_i|/lr)$, being st_i a fixed stop time at node i and lr the cargo load/unload rate. The transportation requests must be fulfilled by a fleet of vehicles with a transport capacity q based on the CD location. Two shipping alternatives are available to fulfill the delivery of any request $\tau\in\Gamma$: shipping directly from the origin $i\in\{I^{+}{}_{\tau}\cap\tau\}$ to the destination $i\in\{I^{-}{}_{\tau}\cap\tau\}$ or shipping from the origin $i\in\{I^{+}{}_{\tau}\cap\tau\}$ to the destination $j\in\{I^{-}{}_{\tau}\cap\tau\}$ via a transshipment operation on the CD. The solution consists of a group of sequences of arcs, called *routes*, such that: (i) for each request, the pick-up activity must precede the delivery task; (ii) each pickup/delivery/mixed route starts and ends on the CD; (iii) each pick-up site $i \in I^+$ is assigned to exactly one route; (iv) each delivery site $i \in I^-$ is assigned to exactly one route; (v) the actual load carried by a vehicle must never exceed the transport capacity q; (vi) the service for any node $i \in I$ must start within the time-window $[a_i, b_i]$; (vii) all activities, including the pick-up tour, the transfer operations and the delivery tours and mixed tours must be completed before a maximum routing time $t^{\max}$ and; (viii) the unload of a cargo from the inbound vehicle on the CD must precede its load on the outbound vehicle. The problem goal is to minimize the total cost for moving the loads from the source sites $i \in I^+$ to the destination sites $i \in I^-$ no matter how.

In order to formulate the problem stated above as an Integer Program (*IP*), let us assume that R^+ denotes the set of pick-up routes, R^- the set of delivery routes and R^{+-} the set of mixed pick-up and delivery routes. For each route $r \in \{R^+ \cup R^- \cup R^{+-}\}$, c_r denote its cost, given by the sum of the costs of the arcs traveled by the vehicle plus a fixed vehicle-utilization-cost. A binary parameter a_{ir} indicates whether route $r \in \{R^+ \cup R^- \cup R^{+-}\}$ visits ($a_{ir} = 1$) or not ($a_{ir} = 0$) the location $i \in I^+ \cup I^-$. The non-negative parameter $t_{ir}{}^+$ indicates, for route r, the availability time on the CD of the request moved from i while the non-negative parameter $t_{jr}{}^-$ is denoting the start time of loading activities on the CD for the load to deliver to site j trough the route r. Parameters $t_{ir}{}^+/t_{ir}{}^-$ are zero if $a_{ir} = 0$. In this model, we use the binary decision variable X_r to determine if the route $r \in \{R^+ \cup R^- \cup R^{+-}\}$ belongs to the optimal solution or not. The problem can now be formulated as:

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

subject to:

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

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

$$\sum_{r \in R^+} t_{ir}^+ X_r \le \sum_{r \in R^-} t_{jr}^- X_r \quad \begin{array}{l} \forall \tau = \{i,j\} \in \Gamma : i \in I^+, i \in I^- \\ X_r = \{0, 1\} \end{array} \tag{4}$$

The objective function (1) minimizes the total routing cost, i.e., the cost of the three kind of routes. Constraints (2) assure that each source site $i \in I^+$ is visited exactly once while constraints (3) guarantee that each destination place $i \in I^-$ is visited exactly once. The constraints (4) are timing constraint assuring that the unloading of a given cargo ends before it is loaded into the delivery truck just in case the cargo is transshipped. So, if a load l_{ij} is moved from its pick-up site to the CD by a route $r \in R^+$ (i.e. $a_{ir} X_r = 1$) and is later delivered to its destination j by a route $r \in R^-$ (i.e. $a_{jr} X_r = 1$), then the cargo must be available in the CD (at a time $t_{ir}{}^+$) before the start-time $t_{jr}{}^-$ of the load activity on the truck that will deliver it.

4. Solution methodology

The model defined by Eqs. (1)–(4) is here embedded into a branch-and-price procedure developed to generate solutions for the problem above formulated. This 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 columns generation approach implicitly handles this complexity by solving the linear relaxation of the formulation (1)–(4), called the reduced master problem (RMP). At the start, an initial but sub-optimal solution is enumerated and the linear relaxation of the RMP is solved considering just this partial set. The solution is then used to determine if there are routes not included in the routes-set that can reduce the objective function value. By using the values of the optimal dual variables of the master constraints with respect to the partial routes-set, new routes are generated and incorporated into the columns pool to re-solve the linear relaxation of the RMP. The procedure iterates between the master problem and the slave routes-generator-problem(s) until no routes with negative reduced costs can be found. Finally, an integer master problem might be solved for finding the best subset of routes. Although the solution found may not be the global optimal, it is usually "close". Because some routes were not generated when solving the relaxed RMP, the procedure is embedded into a branch-and-bound algorithm. This process is named branch-and-price and involves the definition of the relaxed RMP, the slave pricing subproblem(s) and the implementation of a branching rule.

4.1. The master problem

The IP formulation (1)–(4) contains a number of binary variables which grows with the size of the pool of feasible routes. In order to compute a lower bound for its objective function value, we relax the integrality condition for variables X_r and consider the integer problem as a RMP. Initially the poll includes a few columns, representing a feasible solution and the cost of these columns is known in advance. The master problem includes the partitioning constraints (2) and (3) and the transfer-constraints (4). So, the relaxed RMP is defined as follows:

$$\text{Minimize} \sum_{r \in R^+} c_r X_r + \sum_{r \in R^-} c_r X_r + \sum_{r \in R^{+-}} c_r X_r \tag{5}$$

subject to:

$$\sum_{r \in R^+} a_{ir} X_r + 0 + \sum_{r \in R^{+-}} a_{ir} X_r \ge 1 \quad \forall i \in I^+ \tag{6}$$

$$0 + \sum_{r \in R^-} a_{ir} X_r + \sum_{r \in R^{+-}} a_{ir} X_r \ge 1 \quad \forall i \in I^- \tag{7}$$

$$\sum_{r \in R^+} t_{ir}^+ X_r - \sum_{r \in R^-} t_{ir}^- X_r + 0 \le 0 \quad \begin{array}{l} \forall \tau = \{i,j\} \in \Gamma : i \in I^+, i \in I^- \\ 0 \le X_r \le 1 \end{array} \tag{8}$$

Routing problems are naturally modeled by a partitioning formulation because each location is visited just once but they can also be formulated as a set covering problem in which each site must be visited at least once. Since visiting a location once results in the less costly feasible subset, an optimal set-covering solution will be also 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 (Barnhart et al., 2000).

Now, if the optimal solution to the relaxed RMP had been found and if π^+, π^- and π^t are the vectors of optimal dual variables values for constraints (6)–(8), respectively, we can transfer them to the slave pricing problems in order to produce more routes that will be useful to later reduce the value of the objective function (5). Therefore, the pricing problems aim to solve the following three objectives:

$$\hat{c}_r^+ = c_r - \sum_{i \in I^+} \pi_i^+ a_{ir} - \sum_{\tau \in \Gamma} \sum_{i \in I^+ : i \cap \tau \ne 0} \pi_i^t t_{ir}^+ \tag{9}$$

$$\hat{c}_r^- = c_r - \sum_{i \in I^-} \pi_i^- a_{ir} + \sum_{\tau \in \Gamma} \sum_{i \in I^- : i \cap \tau \ne 0} \pi_i^t t_{ir}^- \tag{10}$$

$$\hat{c}_r^{+-} = c_r - \sum_{i \in I^+} \pi_i^+ a_{ir} - \sum_{i \in I^-} \pi_i^- a_{ir} \tag{11}$$

Eqs. (9)–(11) define three subproblems, one for each route type. Eq. (9) is the objective for the search of new pick-up routes, Eq. (10) is the objective for finding new delivery routes and Eq. (11) for new mixed routes. At each iteration of the column generation algorithm the linear relaxation of the RMP is first solved. Then, by using the dual variables in the corresponding pricing problems, new routes (or columns) with negative reduced costs can be found.

4.2. The pricing problems

Each feasible tour is an elementary path from the CD trough some locations $i \in \{I^+ \cup I^-\}$ and back to the CD. The pricing problems are elementary shortest path problems with resource constraints and they are NP-hard in the strong sense. The most used technique to solve the pricing problems was dynamic programming in which a relaxation of the pricing algorithm was solved and cycles were allowed. Nevertheless, the recent trend relies in algorithms in which pricing problems are solved exactly without allowing cycles. Algorithms providing elementary routes may be classified in dynamic programming procedures (Feillet et al., 2004; Chabrier, 2006), mixed integer-linear (MILP) formulations (Dondo, 2012) and constraint-programming algorithms (Gualandi and Malucelli, 2013). In our application we solve exactly the MILP formulation of the elementary pricing problems with a branch-and-cut solver trying to generate many different solutions per iteration.

4.2.1. The slave pick-up problem

The objective of the slave problem for the pickup stage is to find a route r minimizing the quantity stated by Eq. (12) and subject to the constraints stated by Eqs. (13)–(21).

$$\text{Minimize} \left(CV + c\sum_{i \in I^+} T_i^+ - \sum_{i \in I^+} \pi_i^+ Y_i - \sum_{\tau \in \Gamma} \sum_{i \in I^+ : \tau \cap i} \pi_\tau T_i^+ \right) \tag{12}$$

subject to

$$D_i \geq d_{CDi} \quad \forall i \in I^+ \tag{13}$$

$$\left\{ \begin{array}{l} D_j \geq D_i + d_{ij} - M_D(1 - S_{ij}) \quad (14.a) \\ D_i \geq D_j + d_{ij} - M_D S_{ij} - M_D(2 - Y_i - Y_j) \quad (14.b) \end{array} \right\} \quad \forall (i,j) \in I^+ : i < j$$

$$CV \geq cf_v + D_i + d_{CDi} - M_C(1 - Y_i) \quad \forall i \in I^+ \tag{15}$$

$$T_i \geq t_{wi} \quad \forall i \in I^+ \tag{16}$$

$$\left\{ \begin{array}{l} T_j \geq T_i + st_i + t_{ij} - M_T(1 - S_{ij}) \quad (17.a) \\ T_i \geq T_j + st_j + t_{ji} - M_T S_{ij} - M_T(2 - Y_i - Y_j) \quad (17.b) \end{array} \right\} \quad \forall (i,j) \in I^+ : i < j$$

$$TV \geq T_i + t_{iCD} - M_T(1 - Y_i) \quad \forall i \in I^+ \tag{18}$$

$$a_i \leq T_i \leq b_i \quad \forall i \in I^+ \tag{19}$$

$$\left\{ \begin{array}{l} T_{end}^+ \geq TV + st_{CD} - M_T(1 - Y_i) \quad (20.a) \\ T_i^+ \geq T_{end}^+ \quad (20.b) \\ T_i^+ \leq M_T Y_i \quad (20.c) \end{array} \right\}$$

$$\sum_{i \in I^+} Y_i l_i \leq q \quad \forall i \in I^+ \tag{21}$$

The objective function (12) is the cost CV of the generated route minus the prices collected on the visited sites and the prices associated to the unloading times on the CD. The second term of the equation states, as secondary objective, the minimization of the pick-up stage makespan. The parameter c is a very small number aimed at making negligible this term with respect to the three remaining ones. This equation is the pricing reformulation of Eq. (9) where the parameter a_{ir} of the master problem becomes the decision variable Y_i of the pricing one and the parameter t_{ir}^+ correspond to the variable T_i^+. The constraint (13) set the minimum distance to reach the site $i \in I^+$ as the distance of going directly from the CD to location i. The constraints (14) and (15) compute the distances traveled to reach the visited sites $i \in I^+$ and the total cost of the generated route respectively. 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 j ($S_{ij} = 1$), according constraints (14.a), the traveled distance up to the 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 (14.b) becomes active. If one or both sites are not allocated to the tour, Eqs. (14.a) and (14.b) become redundant. M_D is an upper bound for variables D_i. Eq. (15) computes the route-cost CV by the addition of the fixed vehicle utilization cost cf_v to the traveled-distance-cost. Since the last visited pick-up location cannot be known before the problem resolution, Eq. (15) 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. (16)–(18) are similar to constraints (13)–(15) 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. Eq. (19) forces the service time on any site $i \in I^+$ to start at a time T_i bounded by the time window $[a_i, t_i]$. Eq. (20.a) adds to the tour time-length a term related to the unload activities on the CD (st_{CD}) to define the variable value T_{end}^+. By (20.b) the unload time T_i^+ for each transported cargo must be equal or larger than T_{end}^+. Due to the second term of the objective function, this equation is satisfied as equality. By (20.c), if $Y_i = 0$, then $T_i^+ = 0$. Eq. (21) is a capacity constraint for the vehicle traveling the designed tour.

4.2.2. The slave delivery problem

The objective of the slave delivery problem is to find a route r minimizing the quantity stated by the objective function given by Eq. (22) and subject to constraints stated by Eqs. (23)–(31). Eq. (22) is the pricing reformulation of Eq. (10) and the constraints are similar to constraints (13)–(21) but they are used to design delivery routes.

$$\text{Minimize}\quad \left(CV - c\sum_{i\in I^+}(t^{\max} - T_i^-) - \sum_{i\in I^-}\pi_i^- Y_i + \sum_{\tau\in\Gamma}\sum_{i\in I^-:\tau\cap i}\pi_\tau T_i^- \right) \tag{22}$$

subject to

$$D_i \ge d_{CDi} \quad \forall i\in I^- \tag{23}$$

$$\left\{ \begin{array}{c} D_j \ge D_i + d_{ij} - M_D(1-S_{ij}) \\ D_i \ge D_j + d_{ij} - M_D S_{ij} - M_D(2 - Y_i - Y_j) \end{array} \right\} \quad \forall i,j\in I^- : i<j \qquad \begin{array}{r}(24.a)\\(24.b)\end{array}$$

$$CV \ge cf_v + D_i + d_{CDi} - M_C(1-Y_i) \quad \forall i\in I^- \tag{25}$$

$$\left\{ \begin{array}{l} T_i^- \le T_{start}^- \\ T_i^- \le t^{\max} Y_i \end{array} \right\} \quad \forall i\in I^- \qquad \begin{array}{r}(26.a)\\(26.b)\end{array}$$

$$T_i \ge T_i^- + st_{CD} + t_{CDi} - M_T(1-Y_i) \quad \forall i\in I^- \tag{27}$$

$$\left\{ \begin{array}{c} T_j \ge T_i + st_i + t_{ij} - M_T(1-S_{ij}) \\ T_i \ge T_j + st_j + t_{ji} - M_T S_{ij} - M_T(2 - Y_i - Y_j) \end{array} \right\} \quad \forall i,j\in I^- : i<j \qquad \begin{array}{r}(28.a)\\(28.b)\end{array}$$

$$TV \ge T_i + d_{iCD} - M_T(1-Y_i) \quad \forall i\in I^- \tag{29}$$

$$a_i \le T_i \le b_i \quad \forall i\in I^- \tag{30}$$

$$TV \le t^{\max} \tag{31}$$

$$\sum_{i\in I^-} Y_i l_i \le q \tag{32}$$

Eq. (26) states the earliest time at which the request destined to site $i \in I^-$ can be loaded on the vehicle. Eq. (26.b) set the variable value $T_i^- = 0$ if site $i \in I^-$ is not visited by the vehicle traveling the designed tour. The parameter $t^{\max}$ indicates the end-time for all activities. Due to the second term of the objective function, aimed at minimizing, as secondary target, the delivery stage makespan, Eq. (26.a) is satisfied as inequality if $Y_i = 1$.

4.2.3. The slave mixed pick-up and delivery problem

The objective of the slave mixed pick-up and delivery problem is to design a route minimizing Eq. (33) subject to constraints (34)–(43). The formulation is similar to previous slave formulations but in this case the vehicles can visit both pick-up and delivery locations. The objective function is the pricing reformulation of Eq. (11) and the term related to load/unload activities on the CD is not included, obviously because there are not transshipments on the CD. According to Eq. (41) all mixed tours must be fulfilled within the $[0,t^{\max}]$ time-span. The constraint (43.a) forces both locations of a request to be serviced by the same vehicle. Since a request load must be picked-up before the unload activity at destination, Eq. (43.b) set the precedence relationship between both request vertexes.

$$\text{Minimize}\quad \left(CV - \sum_{i\in I^+}\pi_i^+ Y_i - \sum_{i\in I^-}\pi_i^- Y_i \right) \tag{33}$$

Subject to

$$D_i \ge d_{CDi} \quad \forall i\in I^+ \tag{34}$$

$$\left\{ \begin{array}{l} D_j \ge D_i + d_{ij} - M_D(1-S_{ij}) - M_D(2 - Y_i - Y_j) \\ D_i \ge D_j + d_{ij} - M_D S_{ij} - M_D(2 - Y_i - Y_j) \end{array} \right\} \quad \forall (i,j)\in I^+ : i<j \qquad \begin{array}{r}(35.a)\\(35.b)\end{array}$$

$$CV \ge cf_v + D_i + d_{iCD} - M_C(1-Y_i) \quad \forall i\in I^+ \tag{36}$$

$$T_i \ge t_{CDi} \quad \forall i\in I^+ \tag{37}$$

$$\left\{ \begin{array}{l} T_j \ge T_i + st_i + t_{ij} - M_T(1-S_{ij}) - M_T(2 - Y_i - Y_j) \\ T_i \ge T_j + st_j + t_{ji} - M_T S_{ij} - M_T(2 - Y_i - Y_j) \end{array} \right\} \quad \forall (i,j)\in I^+ : i<j \qquad \begin{array}{r}(38.a)\\(38.b)\end{array}$$

$$TV \ge T_i + t_{iCD} - M_T(1-Y_i) \quad \forall i\in I^+ \tag{39}$$

$$t_i^{\min} \le T_i \le t_i^{\max} \quad \forall i\in I^+ \tag{40}$$

$$TV \le t^{\max} \tag{41}$$

$$\sum_{i\in I^+} Y_i l_i \le q \tag{42}$$

$$\left\{ \begin{array}{l} Y_i = Y_j \\ S_{ij} = 1 \end{array} \right\} \qquad \begin{array}{r}(43.a)\\(43.b)\end{array}$$

4.3. The branching strategy

The linear relaxation of the RMP may not return an integer solution and applying a standard branch-and-bound procedure with a given pool of columns may not guarantee an optimal solution (Barnhart et al., 2000). Also a column pricing favorably may exist but it may not be present in the RMP. Consequently, to find the optimal solution, columns must be generated after branching. Therefore, we adopted the Ryan and Foster rule as the branching rule. Ryan and Foster (1981) proposed a branching strategy that is quite popular in column generation + branch-and-price 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 i and j are serviced by the same vehicle and the other where they 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. We implemented this branching scheme by branching on the assignment variables Y_i expect for the combination $Y_i = 0$ for all $i \in I^+ \cup I^-$.

4.4. Implementation issues

The branch-and-price algorithm has been coded in GAMS 23.6.2 by integrating a CG routine into a branch-and-bound procedure. The cores of both algorithms are based on the routines of Kalvelagen (2003a,b). Minor branching and assembling modifications were introduced. They were aimed at replacing the NLP of the Kalvelagen (2003b) MINLP algorithm by the Kalvelagen (2003a) CG procedure and aimed at forbidding the branching combination $Y_i = 0$ for all $i \in I^+ \cup I^-$. The algorithm uses the CPLEX 12 as the MILP sub-algorithm for generating columns and for computing upper and lower bounds. The algorithm runs in a 2.8-Ghz 16-Mbytes RAM PC. Since branch-and-price is an enumeration algorithm enhanced by fathoming based on bound comparisons, it is the best to work with the strongest bounds although the mechanism can work with any bound. This leads to a trade-off between the CPU time used in computing strong bounds and the size of the tree. To reduce the "tailing-off" effect consisting in a very low convergence-rate at the last iterations of the master–slave recursion we ended it after five iterations in no-root nodes and used the bounds computed in such a way. Time-windows reduction and pre-processing were also fully used in order to increase the resolution efficiency. See Dondo (2012) for details on pre-processing and time-windows reduction rules tailored to a slave formulation similar to the ones defined in Section 4.2. The maximum number of nodes to inspect in the incomplete branch-and-price tree is 100 and the columns pool can store up to 10,000 routes. The CPLEX option *solnpool* (CPLEX 12 Solver Manual, 2012) was activated to generate multiple columns per master–slave iteration. The maximum allowed CPU time per master–slave iteration was set to 30 s. Since the maximum number of nodes to inspect is bounded and the master–slave recursion is terminated after five iterations in no-root nodes of the tree, this incomplete procedure is of a heuristic nature. The algorithm can be turned on an exact one by simply removing these limits. To provide an initial solution, feasible routes CD–i–CD are generated for each site $i \in I^+ \cup I^-$. From this initial routes package, the linear RMP can compute the bounds to start the master–slave recursion. 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 three slave tour-generator problems. The master–slaves structure is recursively solved until no more feasible routes can be generated. In such a case, the RMP is solved again to verify the solution integrality. If the solution to this problem is integer, the optimal solution to the original problem has been found and the procedure ends. Otherwise; the integer solution to the RMP or global upper bound (GUB) will have a value higher than the value of the solution to the relaxed RMP or global lower bound (GLB). In that case, 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, the node is fathomed; otherwise it is divided into two child-nodes that are included in the database of unsolved subspaces. Afterwards, 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 Fig. 2.

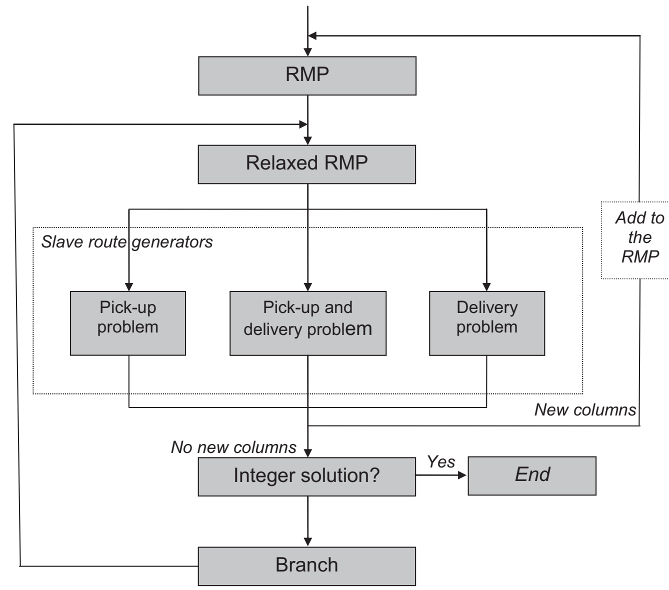


Fig. 2. Outline of the branch-and-price algorithm.

5. Numerical results and a case study

The solution procedure was first tested on a set of instances proposed by Dondo and Cerdá (2013) for solving the VRPCD. The testing allows us to evaluate the performance of the branch-and-price algorithm and to estimate an approximated measure of the quality of the provided solutions in different size instances. A sensibility analysis aimed at testing the influence of parameters changes on the quality of solutions and on the computing time is also carried out. Later the procedure was used to solve four instances of a motivating case study.

5.1. Testing examples

The algorithm was tested with the instances-set proposed by Dondo and Cerdá (2013) These examples define a number of pick-up and delivery sites whose locations are defined by the (X, Y) coordinates in the Euclidean plane. Furthermore, the travel-time between locations is numerically equal to the Euclidean distance. The first half of locations are considered pick-up nodes while the second half are considered delivery locations. The sites are randomly placed in different geographical sites. In this way, the first request is defined by the first pick-up location coupled with the first delivery site. The cargo uploaded on the former site must move to the later one after drop-off/ship-on operations in the CD. The CD is the base of an unspecified number of vehicles, each of them with a cargo capacity $q = 75$ units. Variants with and without time windows were proposed and solved. The examples consider

Table 1
Solutions overview for instances with time windows and the mixed-trips option disabled.

n	Integer solution	Linear solution	Gap (%)	Columns	Total CPU time (s)	Pick-up routes	Delivery routes	Previous Obj. func	Solution CPU time*
10	438.1	435.3	0.73	724	8.8	3	3	438.1	21.5
11	462.3	458.0	1.02	677	8.7	3	3	462.3	8.2
12	505.2	504.7	0.09	643	9.2	3	3	505.2	108.0
13	534.3	534.3	0.00	642	9.1	3	3	534.3	28.0
14	589.3	581.9	1.26	1254	23.6	3	3	597.2	28.8
15	626.4	612.1	2.28	1039	21.4	4	4	628.7	192.4
16	663.1	659.0	0.63	818	20.2	4	4	N.A.	–**
17	685.1	679.2	0.86	964	32.8	4	4	N.A.	–**
18	713.5	698.6	2.09	953	35.3	4	4	N.A.	–**
19	766.1	748.7	2.27	875	30.7	4	5	834.0	212.6
20	808.7	788.2	2.54	1079	69.8	4	5	N.A.	–**
21	831.9	817.6	1.75	1162	157.8	5	5	889.3	148.8
22	864.4	846.2	2.10	1249	170.5	5	5	N.A.	–**
23	914.6	908.5	0.66	983	160.7	5	5	961.6	393.4
24	934.5	924.9	1.02	953	124.2	5	6	N.A.	–**
25	953.4	936.8	1.78	1261	222.4	5	6	1068.2	1827.2
26	964.0	949.9	1.46	1524	282.8	6	6	N.A.	–**
28	968.8	966.5	0.23	1603	396.4	6	6	1045.8	1633.1
30	1016.3	994.1	2.18	1608	1620.5	6	6	1191.5	1065.1
35	1115.1	1062.3	4.73	2079	3938.1	7	8	1183.2	397.8
40	1337.4	1301.8	2.66	2336	3184.4	8	8	1474.0	1158.4
45	1469.7	1435.5	2.33	3059	4376.7	9	9	1544.2	3600**
50	1655.1	1618.4	2.21	3188	5771.9	10	11	1777.4	1183.3

* CPU seconds in a 2.66 MHz six-core dual processor PC with 24 MB RAM.
** Time limit reached.
N.A. not available.

Table 2
Solutions overview for instances without time windows and the mixed-trips option disabled.

n	Integer solution	Linear solution	Gap (%)	Columns	Total CPU time (s)	Pick-up routes	Delivery routes	Previous Obj. func	Solution CPU time (s)*
10	404.1	403.1	0.27	996	27.1	2	2	404.1	105.3
11	414.3	410.3	0.97	994	26.1	2	2	414.3	889.7
12	477.1	456.9	4.22	1228	31.2	3	3	479.8	10,800**
13	515.7	499.7	3.10	1091	36.9	3	3	513.9	3600**
14	554.1	542.9	2.01	1057	46.8	3	3	551.6	3600**
15	589.5	568.4	3.57	1270	123.9	3	3	589.9	3600**
16	613.8	600.3	2.20	1856	298.6	4	4	N.A.	–**
18	664.8	648.8	2.40	1187	574.3	4	4	N.A.	–**
19	697.1	687.1	1.42	1425	738.4	4	4	708.2	3600**
20	723.4	710.0	1.85	1813	1713.4	4	4	N.A.	–**
21	735.8	734.2	0.22	2566	2404.8	4	4	750.2	3600**
22	774.6	765.7	1.14	2225	2404.5	5	5	N.A.	–**
23	866.4	835.6	3.56	2182	1479.4	5	5	883.4	3600**
24	903.6	848.9	6.05	2144	2610.3	5	6	N.A.	–**
25	888.5	856.7	3.58	2181	2963.5	5	5	893.8	3600**
26	890.5	876.8	1.55	1951	2804.2	5	5	N.A.	–**
28	922.4	899.2	2.52	3438	6540.7	6	6	995.6	3600**
30	945.9	927.1	1.99	2512	4012.7	6	6	1020.1	3600**
35	1061.1	1017.9	4.07	3366	5861.7	7	7	1134.1	3600**
40	1287.3	1242.7	3.46	3888	6636.2	8	8	1360.1	3600**
45	1448.9	1395.0	3.72	4752	7102.9	9	9	1522.1	3600**
50	1679.0	1604.5	4.43	6202	11,964.1	10	11	1722.1	3600**

* CPU seconds in a 2.66 MHz six-core dual processor PC with 24 MB RAM.
** Time limit reached.
N.A. Not available

loading/unloading times at pick-up/delivery locations and the CD. The service time at supply and delivery nodes is the sum of two components, a fixed part and a variable service time, with the later one directly increasing with the size of the cargo to be loaded or unloaded. All the problem data can be found in Dondo and Cerdá (2013). These instances are here defined by the number of requests n and the use (or not) of time-windows. The minimum distance solutions to the VRPCD instances with time windows are presented in Table 1 while the solutions without considering time windows are summarized in Table 2.

In order to compare the obtained solutions with regards to solutions previously reported, the mixed-tours option was disabled. Tables 1 and 2 report the integer solution, the lower bound, the duality gap, the total number of generated columns, the consumed CPU time and the number of pick-up and delivery trips specified by the solution, respectively. Previous solution-data reported by Dondo and Cerdá (2013) are also presented. From Tables 1 and 2 it follows that the average duality gap for instances without time windows is slightly larger (2.65%) than the gap for the same instances with time windows (1.61%). The CPU time was, predictably, larger in all examples without time windows. The decomposition algorithm usually found better solutions and is faster than the Dondo and Cerdá's hybrid approach. Although their monolithic formulation was not aimed at proving optimality

Table 3
Solutions overview for instances with time windows and the mixed-trips option enabled.

n	Integer solution	Linear solution	Gap (%)	Columns	Total CPU time (s)	Pick-up routes	Delivery routes	Mixed routes	Savings (%)
10	434.0	430.6	1.01	1037	17.8	0	0	3	0.94
11	454.8	453.3	0.34	1020	18.9	2	2	1	1.65
12	497.3	497.3	0.00	1002	20.3	2	3	1	1.58
13	534.3	534.0	0.00	1281	25.1	3	3	0	0.00
14	589.3	581.9	1.26	1310	32.9	3	3	0	0.00
15	624.2	612.1	1.94	1207	39.9	4	4	0	0.35(*)
16	663.1	659.0	0.63	1316	45.0	4	4	0	0.00
17	685.1	679.2	0.86	1537	68.7	4	4	0	0.00
18	713.5	698.6	2.24	1448	75.7	4	4	0	0.00
19	763.4	748.7	1.92	1546	99.2	4	5	0	0.35(*)
20	806.4	788.2	2.27	2113	95.3	4	5	0	0.28(*)
21	831.9	817.1	1.80	2387	147.2	4	5	0	0.00
22	859.7	846.2	1.57	2268	189.9	5	5	0	0.55(*)
23	918.2	908.5	1.05	2277	176.6	5	5	0	–
24	934.5	924.9	1.02	2191	298.1	5	6	0	0.00
25	954.5	936.2	1.91	2202	1496.3	6	6	0	–
26	959.7	949.5	1.07	2370	2050.0	6	6	0	0.45
28	968.8	966.5	0.23	2721	2702.7	6	6	0	0.00
30	1008.4	994.1	1.42	2834	5504.5	6	6	0	0.78(*)
35	1119.3	1060.4	5.26	2236	10,115.8	7	9	0	–
40	1368.5	1301.9	4.87	2725	11,980.4	8	9	0	–
45	1500.3	1435.6	4.31	2637	10,405.2	9	10	0	–
50	1680.2	1619.6	3.61	2987	12,629.6	10	10	0	–

* A better VRPCD-like solution found.

Table 4
Solutions overview for instances without time windows and the mixed-trips option enabled.

n	Integer solution	Linear solution	Gap (%)	Columns	Total CPU time (s)	Pick-up routes	Delivery routes	Mixed routes	Savings
10	362.5	362.5	0.00	786	61.1	0	0	2	11.47
11	370.2	365.1	1.37	1165	411.8	0	0	2	11.64
12	426.4	399.9	6.23	2261	1222.8	0	0	3	11.89
13	484.6	449.2	9.19	2678	1091.2	0	0	3	6.42
14	514.4	495.1	3.75	2032	908.6	1	1	2	7.78
15	520.1	520.1	0.00	2192	1215.7	0	0	3	13.34
16	586.3	559.6	4.54	3172	2225.3	1	1	3	4.69
18	620.5	601.8	3.10	2943	2685.0	2	2	2	7.13
19	671.9	643.2	4.26	2868	2894.3	2	2	2	3.75
20	702.1	668.1	5.09	3655	4475.4	2	2	2	3.03
21	735.8	720.6	2.06	3127	4223.1	4	4	0	0.00
22	774.5	764.9	1.25	2871	4211.5	5	5	0	0.06
23	852.6	827.3	2.96	3402	6364.2	2	3	3	1.62
24	865.2	846.1	2.26	2530	6199.5	5	5	0	4.44(*)
25	888.5	856.7	3.58	2789	7329.9	5	5	0	0.00
26	890.4	874.8	1.55	1883	4969.5	5	5	0	0.00
28	917.0	899.3	1.93	2413	5447.2	6	6	0	0.59
30	952.6	925.8	2.81	4263	8913.3	6	6	0	–
35	1079.3	1023.3	5.45	6245	14,710.8	8	8	0	–
40	1364.1	1261.4	7.53	4465	6500.2	8	9	0	–
45	1541.6	1433.8	6.99	3998	13,379.5	9	10	0	–
50	1793.6	1629.7	9.14	6960	18,894.9	12	13	0	–

* A better VRPCD-like solution found.

(the search was interrupted after a stated time-length, 3600 s and 10800 s), the authors solved some small instances (n = 10, 11, 12, 13) to proven optimality. In these examples, the monolithic formulation consumed considerably more CPU time than the decomposition algorithm. Tables 3 and 4 report the solution to the same instances but in this case the mixed-tours option is enabled. From this table it is concluded that the saving provided by using mixed tours range from 0% (no costs can be saved by using mixed trips) to 13.34%. Predictably, the CPU time consumed is considerably higher than the CPU consumed by the algorithm adjusted to solve the problem without generating mixed trips. The average duality gap for instances without time windows was also larger (3.87%) than the gap for the same instances with time windows (1.76%). It is worth noting that solution topologies range from solutions with pure pick-up and delivery tours to solutions with just mixed routes. Topologies using the three types of routes were also found. So, optimal and near optimal solutions may have very different topologies.

In large-size examples, the use of mixed trips increases significantly the computational time needed to generate good solutions because the search space is much larger. Moreover, integer solutions reported are worse than the ones reported in Tables 1 and 2. This means that a deeper search, at a higher computational cost, is needed to solve these instances. In order to perform a sensitivity analysis on relevant operational parameters, we re-solved some instances but we changed the vehicles capacity q and the routing time t^{max} because both parameters strongly affect the routes length. Results are summarized in Tables 5–8 and depicted in Fig. 3.

We refrained to solve instances with t^{max} = 300 and time windows because there are some incompatibilities between these windows and the t^{max} value. From the information presented in the tables it follows that the gap remains below the 11% threshold even in the hardest instances; i.e. instances with large vehicles capacities, long routing times and mixed trips. Although the quality of solutions is not very sensitive, the CPU was quite sensitive

Table 5
Solutions overview for instances with time windows and different values of parameters q and t^{max} (mixed trips option disabled).

n	IS	LS	Gap (%)	Columns	CPU time (s)	Pickup routes	Delivery routes
$q=60$; $t^{max}=400$							
30	1119.4	1085.6	3.02	1942	710.1	7	8
35	1217.7	1182.8	2.87	2625	1597.8	8	9
40	1458.0	1440.6	1.19	2817	2275.6	10	10
45	1652.2	1615.0	2.26	2963	2155.9	11	11
50	1928.9	1843.4	4.43	3607	4641.8	12	12
$q=90$; $t^{max}=400$							
30	965.5	941.1	2.53	2578	2792.8	5	6
35	1065.2	1065.2	5.56	2447	4863.1	6	7
40	1290.2	1290.2	4.40	2740	5312.6	7	7
45	1447.9	1447.9	6.31	2064	6579.4	10	8
50	1608.4	1608.4	5.59	2666	6572.8	9	10

Table 6
Solutions overview for instances without time windows and different values of parameters q and t^{max} (mixed trips option disabled).

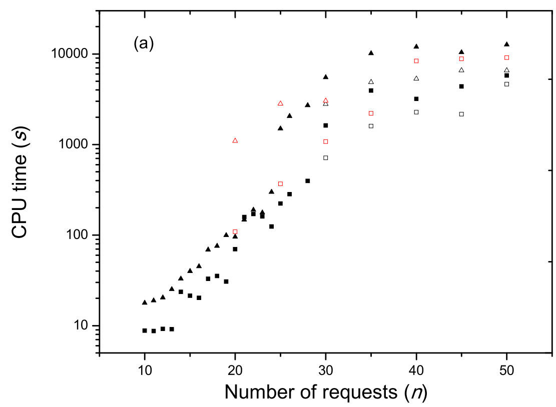
n	IS	LS	Gap (%)	Columns	CPU time (s)	Pickup routes	Delivery routes
$q=60$; $t^{max}=400$							
30	1134.4	1054.6	7.06	1189	2670.0	8	9
35	1224.2	1161.1	5.16	1394	3403.4	9	9
40	1549.3	1418.0	2.83	1949	4764.7	10	10
45	1679.5	1596.2	4.96	2237	5260.9	11	12
50	1875.5	1836.1	2.10	2283	5598.4	12	13
$q=90$; $t^{max}=400$							
30	969.1	872.2	10.0	3714	6679.1	6	6
35	1076.2	965.5	10.3	5373	9046.6	7	8
40	1351.7	1205.6	10.8	6275	11,630.6	9	9
$q=75$; $t^{max}=300$							
20	726.9	710.0	2.32	1054	2162.4	4	4
25	914.3	856.7	6.30	1849	5109.5	5	6
30	988.4	926.4	6.28	3119	5972.6	6	7

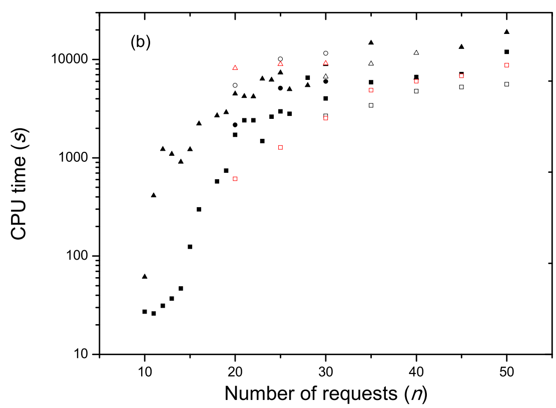
Table 7
Solutions overview for instances with time windows and different values of parameters q and t^{max} (mixed trips option enabled).

n	IS	LS	Gap (%)	Columns	CPU time (s)	Pickup routes	Delivery routes	Mixed routes
$q=60$; $t^{max}=400$								
20	842.3	823.8	2.19	407	108.9	5	5	0
25	1069.6	1009.3	5.64	532	367.5	6	6	1
30	1131.7	1084.7	4.15	784	1079.3	8	8	0
35	1229.7	1181.8	3.90	1288	2208.3	9	9	0
40	1491.8	1440.2	3.46	1314	8348.0	10	11	0
45	1658.9	1614.2	2.69	1376	8850.0	11	12	0
50	1939.5	1842.8	4.99	1592	9098.5	12	13	1
$q=90$; $t^{max}=400$								
20	802.0	778.2	2.89	588	1091.4	5	5	0
25	961.3	902.2	6.15	893	2810.6	7	5	0
30	1021.6	953.9	6.63	1137	3031.4	6	7	0

Table 8
Solutions overview for instances without time windows and different values of parameters q and t^{max} (mixed trips option enabled).

n	IS	LS	Gap (%)	Columns	CPU time (s)	Pickup routes	Delivery routes	Mixed routes
$q=60$; $t^{max}=400$								
20	803.1	774.3	3.59	909	610.6	3	3	2
25	1019.5	965.6	5.29	1088	1275.3	5	6	1
30	1109.3	1053.2	5.06	1656	2536.4	6	7	1
35	1216.2	1160.5	4.58	2252	4881.3	8	8	1
40	1459.5	1411.6	3.28	2264	6002.6	10	10	0
45	1655.9	1594.3	3.72	2401	6811.1	11	11	0
50	1950.2	1822.8	6.53	2982	8761.7	11	12	1
$q=90$; $t^{max}=400$								
20	634.1	631.2	0.45	1810	8164.2	2	2	2
25	811.5	802.4	1.13	1848	8950.9	4	5	0
30	946.4	867.2	8.36	2898	9114.3	5	6	0
$q=75$; $t^{max}=300$								
20	723.5	672.0	7.11	1430	5443.2	1	1	3
25	939.4	856.0	8.81	1728	10,113.2	7	5	0
30	1026.5	925.4	9.87	2672	11,561.8	7	7	0





Symbol	q	t^{max}	Mixed trips option
■	75	400	Disabled
▲	75	400	Enabled
□	60	400	Disabled
Δ	90	400	Disabled
□ (red)	60	400	Enabled
Δ (red)	90	400	Enabled
●	75	300	Disabled
○	75	300	Enabled

Fig. 3. CPU time as a function on the number of requests for different q and t^{max} values on instances with (a) and without (b) time windows.

to parameter variations as it can be observed in Fig. 3. This figure depicts the CPU time as a function on the number of requests for different combinations of values for parameters q and t^{max}.

As expected, instances featuring large vehicle capacities were the hardest to solve, but interestingly, the reduction of t^{max} from 400 to 300 time units did not reduced the CPU time and, on the contrary, instances with short routing horizons seems harder to solve. The explanation lies, likely, in the fact that pick-up and delivery routes must be 'assembled' in a tighter way because there is less time left to perform transshipment operations.

5.2. The motivating case study

After the extensive testing of the previous section, we apply now the solution procedure to the motivating case study next described. A transportation company from Santa Fe (Argentina) provided us with real daily operational data about the distribution of non-perishable products to several industrial (woodworking companies, food companies, small-scale industries) and service companies (supermarkets, retailers) in the Santa Fe urban area. The daily operation here considered involves the use of several vans based on the hub used to transship cargo. Vans are used to collect/deliver small cargo and their maximum volumetric capacity is $q = 7.5\,m^3$. Service times at pick-up/delivery sites are considered approximately constant, $st_i = 30'$ (minutes), and the average urban-travel speed is quite difficult to estimate but is conservatively assumed to be 20 km/h. The case study uses data from a typical working day and involves the fulfillment of 39 transportation requests within this day. We estimated the distance (in km) between customers' locations and between these locations and the CD facility by using the Manhattan distance formula. The datasheet for the instance is detailed in Appendix. Usually the company performs pickup activities during morning and delivery during afternoon to allow transshipments tasks between both stages and to avoid cargo warehousing on depot at night. A fixed van utilization cost cf_v = \$ 100 and a unit distance cost \$10/km are here considered. The fixed cost is doubled for mixed trips because drivers receive higher incomes due to larger and more complex tours. Several requests share an origin location because they represent the movement of goods from a beer factory to

Table 9
Data on solution to the four configurations of the case study.

Option	Integer solution	Linear solution	Columns	Total CPU time (s)	Pick-up routes	Delivery routes	Mixed routes
Without time windows ($t^{max}=600'$)							
VRPCD	3862	3568	2580	4628.6	8	8	–
PDPCD	3450	3287	3704	5082.7	0	0	8
With time windows ($t^{max}=720'$)							
VRPCD	3886	357.1	2282	4289.7	8	8	–
PDPCD	3442	3385	2904	6661.8	2	2	7

Table 10
The solution to the case study with the mixed trips option disabled (no time windows considered).

Pick-up stage				
Pick-up route	Tour	Load collected (m^3)	Route time-length ($'$)	Route cost ($)
1	CD-r28-r24-r22-r10-CD	7.3	168.9	26.3
2	CD-14-r33-r18-r15-r17-CD	7.5	199.1	26.4
3	CD-r29-r23-r32-r34-CD	7.5	197.5	35.8
4	CD-r8-r43-r44-r36-CD	7.1	155.6	21.9
5	CD-r20-r7-r6-r5-r4-r3-CD	7.5	238.1	22.7
6	CD-r13-r42-r12-r31-r9-CD	7.5	180.0	20.0
7	CD-r35-r21-r19-r41-r40-CD	7.5	205.6	18.5
8	CD-r39-r38-r37-r30-r11-CD	7.4	172.1	17.4

Timelines at the cross-dock					
Unloading tasks		Loading tasks			
Pick-up route	End-unloading time ($'$)	Delivery route	Last request available	Availability time ($'$)	Start-loading time ($'$)
1	198.9	1	r23	227.5	390.9
2	229.1	2	r7	268.1	379.0
3	227.5	3	r5	268.1	401.0
4	185.6	4	r6	268.1	454.9
5	268.1	5	r4	268.1	337.8
6	210.0	6	r3	268.1	334.7
7	235.6	7	r34	227.5	412.0
8	202.1	8	r20	268.1	320.7

Route	Load to deliver (m^3)	Departure time ($'$)	Tour (m^3)	Route cost ($)
1	7.5	420.9	CD-r37-r39-r30-r23-CD	230
2	7.5	409.0	CD-r7-r32-r44-r13-r8-CD	237
3	7.5	431.0	CD-r22-r11-r10-r5-CD	264
4	7.5	484.9	CD-r33-r43-r6-CD	184
5	7.2	367.8	CD-r35-r41-r12-r4-r9-r38-CD	274
6	7.3	364.7	CD-r24-r31-r18-r21-r14-r3-CD	284
7	7.3	442.0	CD-r36-r28-r34-r42-CD	227

Fixed costs	$ 1600
Routing costs	$ 2262
Total costs	$ 3862
Routing time	3006.4$'$
Cross-docking time	480.0$'$

Table 11
The solution to the case study with the mixed trips option enabled (no time windows considered).

Mixed routes				
Terminal	Tour	Maximum onboard load (m^3)	Route cost ($)	Return time ($'$)
1	CD-r8^{+}-r36^{+}-r22^{+}-r5^{+}-r22^{-}-r36^{-}-r13^{+}-r5^{-}-r8^{-}-r13^{-}-CD	6.1	53.1	399.4
2	CD-r11^{+}-r30^{+}-r39^{+}-r38^{+}-r17^{+}-r30^{-}-r17^{-}-r39^{-}-r38^{-}-r11^{-}-CD	7.4	38.5	355.3
3	CD-r37^{+}-r23^{+}-r24^{+}-r28^{+}-r24^{-}-r23^{-}-r28^{-}-r37^{-}-CD	7.5	44.5	313.4
4	CD-r14^{+}-r16^{+}-r21^{+}-r20^{+}-r44^{+}-r20^{-}-r16^{-}-r14^{-}-r21^{-}-r44^{-}-CD	7.4	37.7	353.2
5	CD-r12^{+}-r42^{+}-r31^{+}-r31^{-}-r4^{+}-r7^{+}-r42^{-}-r12^{-}-r4^{-}-r7^{-}-CD	5.9	48.3	385.0
6	CD-r9^{+}-r29^{+}-r10^{+}-r29^{-}-r19^{+}-r10^{-}-r34^{+}-r9-r34^{-}-r19^{-}-r43^{+}-r43^{-}-CD	3.5	43.4	430.2
7	CD-r33^{+}-r32^{+}-r33^{-}-r18^{+}-r18^{-}-r32^{-}-CD	7.0	45.5	256.4
8	CD-r35^{+}-r40^{+}-r41^{+}-r35^{-}-r41^{-}-r15^{+}-r6^{+}-r3^{+}-r15^{-}-r40^{-}-r3^{-}-r6^{-}-CD	5.1	34.0	402.1
Fixed costs				$ 1600
Routing costs				$ 1850
Total costs				$ 3450
Routing time				2895.0$'$
Cross-docking time				–

Table 12
The solution to the case study with the mixed trips option disabled (time windows considered).

Pick-up stage				
Pick-up route	Tour	Load collected (m^3)	Route time-length (′)	Route cost ($)
1	CD-r21-r7-r4-r3-r43-r8-CD	7.5	218.3	22.8
2	CD-r40-r19-r20-r22-r10-r37-CD	7.5	237.5	29.2
3	CD-r39-r33-CD	7.0	168.6	26.0
4	CD-r41-r28-r24-r23-CD	7.5	175.1	28.4
5	CD-r29-r31-r42-r12-r13-r14-CD	7.5	212.8	20.0
6	CD-r38-r36-r17-r30-r11r9-CD	7.4	213.0	17.6
7	CD-r35-r34-r32-r16-CD	7.5	228.6	23.1
8	CD-r15-r6-r5-r18-r44-CD	7.4	188.6	22.9

Timelines at the cross-dock					
Unloading tasks		Loading tasks			
Pick-up route	End-unloading time (′)	Delivery route	Last request available	Availability time (′)	Start-loading time (′)
1	248.3	1	r34	258.6	493.8
2	267.5	2	r40	267.5	484.1
3	198.6	3	r10	267.5	523.3
4	205.1	4	n31	242.8	456.6
5	242.8	5	n19, n20	267.5	422.5
6	243.0	6	n37	267.5	513.3
7	258.6	7	n22	267.5	522.9
8	218.6	8	n32	258.6	498.2

Route	Load to deliver (m^3)	Departure time (′)	Tour (m^3)	Route cost ($)
1	7.3	523.8	CD-r9-r11-r4-r12-r34-CD	25.4
2	7.2	504.1	CD-r24-r3-r40-r17-r36-CD	28.6
3	7.4	543.3	CD-r10-r43-r14-r44-CD	25.6
4	7.5	476.6	CD-r28-r38-r39-r31-r29-CD	25.6
5	7.5	442.5	CD-r21-r18-r30-r20-r19-r15-r16-CD	29.2
6	7.4	533.3	CD-r7-r33-r37-r41-r35-CD	18.9
7	7.5	542.9	CD-r42-r22-r23-CD	21.4
8	7.5	518.2	CD-r6-r32-r13-r8-r5-CD	23.9

Fixed costs	$ 1600
Routing costs	$ 2286
Total costs	$ 3886
Routing time	3317.8′
Cross-docking time	480.0′

clients. Also some requests share an end point because reverse logistics transportation of some empty pallets. We evaluated four consolidation and distribution configurations. The first configuration involves just pick-up and delivery tours and can be named as the 'VRPCD option'. The second one allows the use of mixed tours and can be labeled as the 'PDPCD option'. The working time length is $t^{max} = 10$ h for both instances. The third and fourth configurations introduce the time-windows specified in Appendix to both previous instances. The working time-length was extended to $t^{max} = 12$ h in both instances to match it with stated time windows corresponding to partitioned commercial working days (morning and afternoon). The results for all configurations are summarized in Table 9. The solutions obtained for all of them are specified in Tables 10–13.

Figs. 4 and 5 illustrate the solution shapes for the instances without mixed trips. Note that in the configuration without time windows and with $t^{max} = 600'$, all requests can be cheaper satisfied with mixed trips without need of transhipment operations. The change of the working time-span to adapt it to the time windows corresponding to a partitioned commercial working day changed substantively the topology of the solutions. Two pick-up tours and

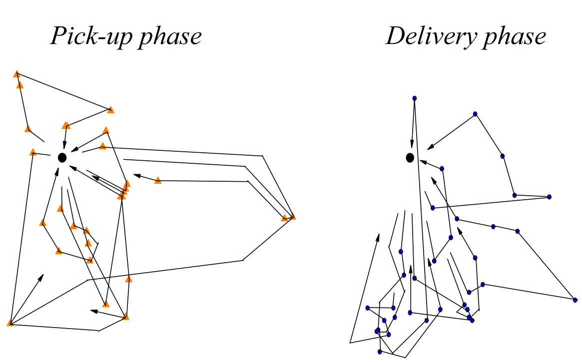


Fig. 4. The VRPCD solution to the case study (no time windows considered).

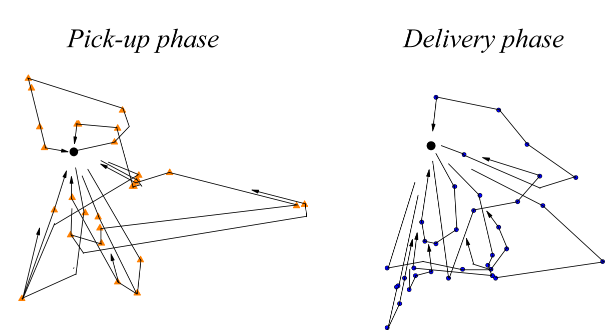


Fig. 5. The VRPCD solution to the case study (time windows considered).

Table 13
The solution to the case study with the mixed trips option enabled (time windows considered).

Pick-up route	Tour	Load collected (m^3)	Route time-length (′)	Route cost ($)
1	CD-r17-r35-CD	2.2	78.9	163
2	CD-r20-r15-r6-r5-r16-r8-CD	7.5	221.9	240

Timelines at the cross-dock

	Unloading tasks	Loading tasks			
Pick-up route	End-unloading time (′)	Delivery Route	Last request available	Availability time (′)	Start-loading time (′)
1	108.9	1	r5-r6	251.9	615.6
2	251.9	2	r20-r15-r16-r8	251.9	421.6

Route	Load to deliver	Departure time	Tour	Tour cost
1	3.0	645.6	CD-r5-r6-CD	148
2	6.7	451.1	CD-r8r35-r20-r16-r17-r15-CD	296

Mixed routes

Terminal	Tour	Maximum onboard load (m^3)	Tour cost ($)	Return time (′)
1	CD-n38-n28-n23-n38-n23-n28-CD	5.7	442	593.9
2	CD-n14-n13-n9-n30-n13-n10-n36-n14-n30-n10-n9-n36-CD	4.1	426	604.9
3	CD-r12-r42-r11-r29-r37-r29-r19-r37-r12-r11-r42-r19-CD	6.6	480	599.0
4	CD-r39-r7-r3-r39-r43-r43-r3-r7-CD	4.7	331	609.4
5	CD-r40-r40-r4-r4-r33-r33-CD	4.0	409	242.6
6	CD-r31-r21-r24-r22-r21-r41-r41-r22-r24-r31-CD	6.7	520	587.4
7	CD-r44-r18-r34-r32-r34-r18-r32-r44-CD	7.2	457	348.1

Fixed costs	$1800
Routing costs	$1642
Total costs	$3442
Routing time	4233.0′
Cross docking time	120′

two delivery tours are involved in the PDPCD option solution. The remaining requests are fulfilled via mixed trips. The solution is slightly cheaper than the VRPCD-like solution detailed in Table 12 and depicted in Fig. 5.

6. Conclusions

A truncated branch-and-price solution-algorithm to solve a realistic problem involving the fulfillment of a list of transportation requests by choosing between two different delivery options has been presented. The problem arises from a logistic company that consolidates and delivers cargo for service and production companies supply-chains. Two shipping alternatives were considered in the problem: a direct delivery to the destination using a single vehicle and a delivery via transshipment at the CD. The problem was first modeled as a set partitioning problem with additional transshipment time-coordinating constraints. The model was then reformulated and embedded into an incomplete branch-and-price solution-mechanism. The proposed mechanism has the following original features: (i) it introduces transshipment time-coordinating constraints into the set-partitioning formulation of the problem; (ii) it utilizes multiple routes-generator problems at the slave level of the CG procedure. Since the problem involves three types of routes, specific integer-linear programs for each routes-type were also developed, and; (iii) the pricing problems were formulated as integer-linear programs and were solved by a branch-and-cut solver trying to maximize the solutions diversification in order to obtain a maximum number of elementary columns per master–slave iteration. A standard branching mechanism was used to explore the bounded branch-and-price tree. The proposed algorithm was validated by first solving numerous academic-type instances involving up to 50 transportation requests. The problems were solved with and without the possibility of using mixed routes in order to compare results and to estimate the saving provided by such a routing option. The comparison showed that sizable costs-savings may be provided by the mixed-tours option in numerous instances. Remarkably, all examples were solved in less than five hours and most of them in less than an hour. Afterwards a sensitivity analysis on variations of the routing time and of the vehicle capacity was performed. It showed that the quality of solutions is fairly good in all instances but also showed that the CPU time is quite sensitive to changes in these parameters. Finally we presented the realistic case study that motivated the development of the model and the solutions found for four variants of this case. A future research to complement this work should consider the following issues: (i) to enlarge the size of solved examples and/or reduce the duality gap, more elaborated and efficient models and methods to provide elementary routes to the RMP may be developed; (ii) a more complex formulation of mixed trips that track the load onboard the vehicles can be used with the aim of a more efficient utilization of the vehicle capacity; (iii) transshipment timing constraints may be introduced in multi cross-docking networks as the ones considered in Dondo and Mendez (2014), and; (iv) the extension to more generalized supply chain problems where the concept of transportation requests is replaced or complemented by sets of cargo-source and cargo-sink locations must be also considered.

Acknowledgements

This work was partially supported by grant 2010-161-12 from Secretaría de Estado de Ciencia, Tecnología e Innovación de la Provincia de Santa Fe. The authors want to thank *Transporte Poccia Hermanos* for supporting our grant application and for providing the daily operational data for the case study.

Appendix A. Appendix

Transportation requests datasheet.

Request	Cargo volume (m^3)	Pick-up				Delivery			
		X_{coord}	Y_{coord}	a_i (′)	b_i (′)	X_{coord}	Y_{coord}	a_i (′)	b_i (′)
n3	1.2	100	55	0	120	92	118	540	600
n4	1.8	100	55	0	120	33	14	0	720
n5	2.0	100	55	0	120	64	195	660	720
n6	1.0	100	55	0	120	82	150	0	720
n7	0.5	100	55	0	120	76	125	0	720
n8	0.3	60	124	0	720	104	185	0	720
n9	1.2	63	177	180	300	40	47	0	720
n10	1.5	120	142	0	720	72	53	0	720
n11	2.0	64	177	0	720	39	46	480	600
n12	1.1	35	203	0	720	41	33	0	720
n13	1.2	40	175	0	720	122	158	0	720
n14	0.9	43	160	180	300	116	113	0	720
n15	1.4	102	79	0	720	100	55	600	720
n16	1.8	49	115	0	720	100	55	600	720
n17	0.7	97	132	0	720	100	55	600	720
n18	0.5	100	55	60	120	170	66	0	720
n19	0.8	78	91	0	720	100	55	540	660
n20	1.0	77	102	0	720	100	55	540	660
n21	1.2	76	110	0	720	132	110	0	720
n22	2.0	198	118	0	720	51	55	0	720
n23	3.0	203	119	60	180	61	58	0	720
n24	2.0	203	119	60	180	100	71	0	720
n28	1.8	203	119	60	180	33	61	0	720
n29	0.3	86	164	0	720	104	93	0	720
n30	0.8	88	174	0	720	102	53	0	720
n31	1.5	91	187	0	720	109	76	540	660
n32	3.0	29	51	120	240	153	132	0	720
n33	4.0	29	51	120	240	77	91	0	720
n34	1.2	29	51	120	240	44	53	0	720
n35	1.5	68	113	0	720	55	97	0	720
n36	1.8	98	132	0	720	50	61	0	720
n37	0.7	99	135	180	300	64	80	0	720
n38	0.9	100	137	0	720	81	60	480	540
n39	3.0	101	140	0	720	99	60	0	720
n40	1.5	59	97	0	720	97	63	0	720
n41	0.7	59	97	0	720	57	82	0	720
n42	2.5	33	210	0	720	47	44	510	570
n43	2.5	88	63	30	150	88	106	0	720
n44	2.5	88	63	30	150	130	133	0	720

Depot coordinates: (X_{coord} = 61; $X_{coord.}$ = 147).
Distances between locations (in km) are estimated as: $d_{ij} = 0.045^{*}(|X^{i}{}_{coord} - X^{j}{}_{coord}| + |Y^{i}{}_{coord} - Y^{j}{}_{coord}|)$.
Travel time between locations (in ′) are estimated as: $t_{ij} = 3d_{ij}$
*t = 0′ corresponds to 8:00 AM. The campaign ends at t = 720′ = 8:00 PM.

References

Ahuja R, Ergun O, Orlin J, Punnen A. A survey of very large scale neighborhood search techniques. Discret Appl Math 2002;123:75–102.

Apte U, Viswanathan S. Effective cross-docking for improving distribution efficiencies. Int J Logist Res Appl 2000;3:291–302.

Ball M, Magnanti T, Monma C, Nemhauser G. Network routing. Amsterdam Elsevier Science; 1995.

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):316–29.

Bodin L, Golden B, Assad A, Ball M. Routing and scheduling of vehicles and crews, the state of the art. Comput Oper Res 1983;10(2):62–212.

Boloori Arabani A, Fatemi Ghomi S, Zandieh M. Meta-heuristics implementation for scheduling of truck in a cross-docking system with temporary storage. Expert Syst Appl 2011;38:1964–79.

Boysen N. Truck scheduling at zero-inventory cross docking terminals. Comput Oper Res 2010;37:32–41.

Boysen N, Fliedner M. Cross dock scheduling: classification, literature review and research agenda. Omega 2010;38:413–22.

Chabrier A. Vehicle routing problem with elementary shortest path based column generation. Comput Oper Res 2006;33(10):2972–90.

CPLEX 12 Solver Manual, 2012.

Desrosiers J, Dumas Y, Solomon M, Soumis F. Time constrained routing and scheduling. In: Ball M, editor. Handbook in OR and MS, vol. 8. Elsevier Science; 1995. p. 35–139.

Dondo R. A new MILP formulation to the shortest path problem with time windows and capacity constraints. Latin Am Appl Res 2012;42:257–65.

Dondo R, Cerdá J. A sweep-heuristic based formulation for the vehicle routing problem with cross docking. Comput Chem Eng 2013;48:293–311.

Dondo R, Mendez C. A branch-and-price approach to manage cargo consolidation and distribution in supply chains. Ind Eng Chem Res 2014;53(44):17226–39.

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):3002–24.

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):216–29.

Grossmann I. Advances in mathematical programming models for enterprise-wide optimization. Comput Chem Eng 2012;47:2–18.

Gualandi S, Malucelli F. Constraint programming-based column generation. Ann Oper Res 2013;204:11–32.

Kalvelagen E. Columns generation with GAMS; 2003a, Downloaded from http://amsterdamoptimization.com/pdf/colgen.pdf

Kalvelagen E. Some MINLP solution algorithms; 2003b, Downloaded from http://www.amsterdamoptimization.com/pdf/minlp.pdf

Lee Y, Jung J, Lee K. Vehicle routing scheduling for cross-docking in the supply chain. Comput Ind Eng 2006;51:247–56.

Li Z, Low M, Shakeri M, Lim Y. Crossdocking planning and scheduling: problems and algorithms. SIMTech Technical Reports 2009;10(3):159–67.

Liao Ch, Lin Y, Shih S. Vehicle routing with cross-docking in the supply chain. Expert Syst Appl 2010;37:6868–73.

Papageorgiou L. Supply chain optimization for the process industries: advances and opportunities. Comput Chem Eng 2009;33:1931–8.

Ryan D, Foster B. An integer programming approach to scheduling. In: Wren A, editor. Computer scheduling of public transport urban passenger vehicle and crew scheduling. North-Holland: Amsterdam; 1981. p. 269–80.

Santos F, Mateus G, da Cunha A. A branch-and-price algorithm for a vehicle routing problem with cross-docking. LAGOS'11 – VI Latin-American algorithms, graphs and optimization symposium. Electronic Notes in Discrete Mathematics, vol. 37; 2011. p. 249–54.

Santos F, Mateus G, da Cunha A. The pickup and delivery problem with cross-docking. Comput Oper Res 2013;40:1085–93.

Tsui L, Chang C. An optimal solution to a dock door assignment problem. Comput Ind Eng 1992;23(1–4):283–6.

Van Belle J, Valckenaers P, Cattrysse D. Cross-docking: state of the art. Omega 2012;40:827–46.

Wen M, Larsen J, Clausen J, Cordeau J, Laporte G. Vehicle routing with cross-docking. J Oper Res Soc 2009;60:1708–18.

Yu W, Egbelu P. Scheduling of inbound & outbound trucks in cross docking systems with temporary storage. Eur J Oper Res 2008;184:377–96.