Cargo consolidation and distribution through a terminals-network. A branch-and-price approach.

Rodolfo Dondo[1]

[1] INTEC (UNL - CONICET) - Güemes 3450 - 3000 Santa Fe – ARGENTINA
Rodolfo Dondo, rdondo@santafe-conicet.gov.ar

Abstract. Less-than-truckload (LTL) is a transport modality that includes many practical variations to convey a number of transportation-requests from the origin locations to their destinations by using the possibility of goods-transshipments on the carrier's terminals-network. In this way logistics companies are required to consolidate shipments from different suppliers in the outbound vehicles at a terminal of the network. We present a methodology for finding near-optimal solutions to a LTL shipping modality used for cargo consolidation and distribution through a terminals-network. The methodology uses column generation combined with an incomplete branch-and-price procedure.

Keywords: cargo consolidation; distribution; less-than-truckload; branch-and-price; transshipment, multiple terminals.

1 Introduction

Suppliers, manufacturers, warehouses and customers are the major components of the so-called supply chain (SC) carrying goods from the upstream 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 supply chain. 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 [2]. In order to operate with high efficiency a LTL system must deal with complex issues like, for example, how truck loading and unloading should be scheduled at the terminals and how vehicles should be routed. The way goods are collected and delivered is of crucial importance for determining the cargo flows and workload on terminals. However, cost-effective shipping is not the only challenge for carriers since they have to ensure a certain service-quality level. This work presents a truncated branch-and-price decomposition-approach to provide solutions to a problem related to the LTL shipping-mode. The solutions consist on a set of pick-up, delivery, pick-up-and-delivery, and transfer routes used to move cargo from the stated source locations to the started destinations. This work builds on a previous one [3] and aims at assembling pick-up, transfer and delivery tours on a flexible way in order to avoid the rigid time delimitation constraints used in such a work.

2 Modeling and Defining the Problem

A LTL carrier operates a terminals-network to provide convey services during a specified time period as, for example, on a daily basis. The company 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 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 terminal is sent to a "break-bulk" terminal where it may be consolidated with freight from other end-of-lines terminals. The terminals-network of the carrier and the cargo-source and destination locations to visit are illustrated in Figure 1.

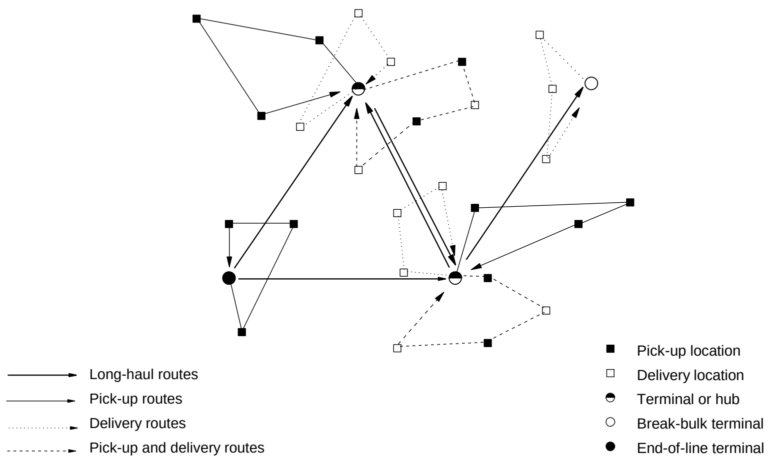

Fig. 1: A typical two-levels network used for cargo consolidation and distribution (reprinted from [3]).

This two-echelons network involves an upper level sub-network connecting terminals and a lower level sub-network connecting source and destiny locations. Vehicles picking and/or delivering cargo travel along the low-level network to bring freight to terminals and to move freight from terminals to destinations. Consolidation at a terminal requires freight to be cross-docked which results in handling costs. Freight transportation between terminals is carried out by the co-called long-haul trucks. So, there are several ways to deliver a shipment: it may be directly moved from its origin to its destination, it may be sent to the terminal serving the area and from there delivered to the destination and; from one terminal it may be sent to another terminal and from there delivered to the destination. The problem is formally defined as follows:

Let the transportation network be represented by a directed graph $G(T \cup \Gamma^+ \cup \Gamma; A)$ comprising a set T of terminals that operate as origin and destination of local and long-haul shipments; a set Γ^+ of pick-up locations and the set Γ of delivery sites. The list of route-arcs connecting them is defined by A. Non-negative values d_{ij} and t_{ij} are associated to each arc $(i,j) \in A$, representing respectively the travel distance/cost and the travel-time to reach the site j starting from the location i. A transportation request $\tau = \{i, j\}$ of a request list $\Gamma = \{ \tau_1, \dots \tau_n\}$ consists of a demand for a transportation service from the origin-location $i \in \{\Gamma^+ \cap \tau\}$ to the destination location $j \in \{\Gamma \cap \tau\}$ for a stated load l_{ij}. Visits must start within stated time windows $[t_i^{min}, t_i^{max}]$ for all pick-up sites $i \in \Gamma^+$ and $[t_j^{min}, t_j^{max}]$ for sites $j \in \Gamma$. These time-windows must also be compatible. Fixed service times st_i are spent at each pickup/delivery location $i \in \{\Gamma^+ \cup \Gamma\}$. The shipping alternatives available to fulfill the delivery of any request $\tau \in \Gamma$ are: (i) Shipping on a local vehicle directly from the origin $i \in (\Gamma^+ \cap \tau)$ to the destination $j \in (\Gamma \cap \tau)$. (ii) Shipping from the origin $i \in (\Gamma^+ \cap \tau)$ to the destination $j \in (\Gamma \cap \tau)$ via cross-docking on a single terminal $t \in T$. (iii) Shipping from the origin $i \in (\Gamma^+ \cap \tau)$ to the destination $j \in (\Gamma \cap \tau)$ 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 trips starts/ends and the long-haul flow between terminals must be determined by the solution. The operational costs depend on the number of pick-up, delivery, pick-up-and-delivery and long-haul routes and on the number of incurred cross-docking operations. The objective is to minimize the sum of cross-docking costs, vehicles fixed costs and traveling costs while satisfying the following operational constraints: (a) All pick-up and delivery sites must be visited just once and only by one vehicle. (b) The service at each customer must start within its time window. (c) Each pick-up/delivery/mixed route begins at a terminal and ends at the same terminal. (d) The sum of the collected/delivered loads in each pick-up/delivery/mixed route must not exceed the capacity of the in-route vehicle. (e) All routes must be fulfilled within the time-interval $[0,t^{max}]$.

In [3] this problem was tackled by partitioning the whole time-horizon $[0,t^{max}]$ in three stages; a pick-up stage bounded by the time-interval $[0,t^{max+}]$, a transfer stage bounded by the interval $[t^{max+}, t^{min-}]$ and a delivery stage bounded by the interval $[t^{min-},t^{max}]$. Furthermore, a request can be directly driven from its origin to its destination by a mixed pick-up-and-delivery trip during the whole time-interval $[0, t^{max+}]$. The rigid time-delimitation imposed to pure pick-up routes and pure delivery-routes lead to a constrained solution space that may exclude good solutions assembling, for example, a “long” pick-up route with a “short” delivery route. So, we propose in this work to drop the hard time delimitation between these steps and let the solution procedure to fix the routes time lengths for routes other than the mixed and transfer routes.

In order to model this problem as an Integer Program (*IP*), let us assume that R^T denotes the set of long-haul routes, R^+ 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 $r \in \{R^T \cup R^+ \cup R^- \cup R^{+-}\}$, c_r denote its cost, given by the sum of the costs of the arcs travelled by the vehicle plus a given fixed vehicle-utilization-cost. Long-haul routes $r \in R^T$ include also the cost of the associated cross-docking operations at start/end terminals. We are also given a binary parameters a_{ir} indicating whether route $r \in \{R^+ \cup R^- \cup R^{+-}\}$ visits ($a_{ir} = 1$) or not ($a_{ir} = 0$) the location $i \in I^+ \cup I^-$. For a route $r \in \{ R^+ \cup R^- \cup R^{+-}\}$, we consider also a binary parameter b_{rt} that assumes value 1 if route r starts/end on the terminal t and 0 otherwise. In that model, we use the binary decision variable X_r to determine if the route $r \in \{R^T \cup R^+ \cup R^- \cup R^{+-}\}$ belongs to the optimal solution or not. The problem can now be formulated as:

Minimize

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

Subjet to

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

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

$$\left\{ \begin{array}{c} \sum_{t\in T} b_{tr}{}^{start} \sum_{r\in R^+} a_{ir} b_{rt} X_r + \sum_{t\in T} b_{tr}{}^{end} \sum_{r\in R^-} a_{ir} b_{rt} X_r - 1 \le X_r \quad (4) \\ \sum_{t\in T} b_{tr}{}^{start} \sum_{r\in R^+} a_{ir} b_{rt} t^+_{r\,end} X_r + X_r t_r^{transf} \le \sum_{t\in T} b_{tr}{}^{end} \sum_{r\in R^-} a_{ir} b_{rt} t^-_{r\,start} X_r \quad (5) \end{array} \right\} \qquad \forall r \in R^T,\ \tau = \{i, j\} \in \Gamma$$

The parameter $t_r{}^+{}_{end}$ stands for the end-time of unload activities for the route $r \in R^+$, $t_r{}^{transf}$ is the transfer time of the long haul route $r \in R^T$ while $t_r{}^-{}_{start}$ is the start time of loading activities for the route $r \in R^-$. The objective function (1) minimizes the cost of all kind of routes. Constraint (2) assures that the source site $i \in I^+$is visited exactly once while constraints (3) guarantee that each destination place $i \in I^-$ is visited exactly once. Inequalities (4) are transfer constraints imposing that long-haul route $r = (t, t')\in R^T$ is used whenever the load picked-up from its source site $i \in I^+$ is unloaded on the terminal t and loaded on the terminal t' for its delivery to the destination site $i \in I^-$. Constraint (5) coordinates in the time dimension these transfers. I.e. it states that the start-time of the route delivering the cargo associated to request τ must be larger than the sum of the transfer-time and the time at which this cargo is unloaded on the start-terminal t of the transfer route (t, t'). Both indexes t and t' may refer to the same physical terminal to consider the shipping option (ii). Since the number of terminals is much smaller than the number of pick-up and delivery locations and because the transfer routes involve a single arc, they can be totally enumerated. It is not possible to generate all feasible routes $r \in \{R^+ \cup R^- \cup R^{+-}\}$ but a column generation approach handles this complexity by implicitly considering all of them through the solution of the linear relaxation of the formulation (1)-(5), 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 considering just 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 routes-generator-problems until no routes with negative reduced costs can be found. After that, an integer master problem may be solved for finding the best subset of routes. The procedure must be embedded into a branch-and-bound algorithm to find the optimal subset because some routes that were not generated when solving the relaxed RMP may be needed to solve the integer one. Finally, the solution is specified by solving, a travelling salesman problem with time windows for each selected column. The process is named branch-and-price and involves the definition of the linear RMP, the definition of the slave routes-generator or pricing problems and the implementation of a branching rule.

2.1 The Master Problem

To obtain the RMP we reorder the constraints (4) and (5) to give rise to the following relaxed RMP:

Minimize (1)

$$\sum_{r\in R^T} c_r X_r + \sum_{r\in R^+} c_r X_r + \sum_{r\in R^-} c_r X_r + \sum_{r\in R^{+-}} c_r X_r$$

subject to:

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

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

$$\left\{ \begin{array}{l} \sum_{r\in R^+} \alpha_{ir}^+ X_r + \sum_{r\in R^-} \alpha_{ir}^- X_r + 0 - X_r \leq 1 \\ \sum_{r\in R^+} \beta_{ir}^+ X_r + \sum_{r\in R^-} \beta_{ir}^- X_r + 0 + \sum_{r\in R^T} \beta_{ir}^T X_r \leq 0 \end{array} \right\} \qquad \forall r \in R^T,\ \tau = \{i, j\} \in \Gamma \qquad (4)\ (5)$$

$$0 \leq X_r \leq 1$$

where

$$\alpha_{ir}^+ = \sum_{t\in T} b_t^{start} b_{rt} a_{ir} \qquad \forall r \in R^T, i \in \left(I^+ \cap r\right) \tag{6.a}$$

$$\alpha_{ir}^{-} = \sum_{t \in T} b_t^{end} b_{rt} a_{ir} \qquad \forall r \in R^T, i \in (I^- \cap r) \qquad (6.b)$$

$$\beta_{ir}^{+} = \sum_{t \in T} b_t^{start} b_{rt} a_{ir} t_{r\,end}^{+} \qquad \forall r \in R^T, i \in (I^+ \cap r) \qquad (6.c)$$

$$\beta_{ir}^{-} = -\sum_{t \in T} b_t^{end} b_{rt} a_{ir} t_{r\,start}^{-} \qquad \forall r \in R^T, i \in (I^- \cap r) \qquad (6.d)$$

$$\beta_{ir}^{T} = t_r^{transf} \qquad \forall r \in R^T \qquad (6.e)$$

The RMP was expressed according the $\boldsymbol{A}x \geq \boldsymbol{b}$ mathematical structure, in which the first column of constraints (5) correspond to all generated pick-up routes, the second column to all generated delivery routes, the third one to the generated mixed routes and the last one to the enumerated transfer routes. The zeros represent missing routes on each block. E.g. the zero in the second column of constraint (2) mean that pure delivery routes can't visit a pick-up site $i \in I^+$. The first three columns arising from eqs. (1) to (5) define the respective pricing problems. The last column is associated to the transfer routes. Since they were pre-enumerated, their generation is not necessary.

2.2 Pricing sub-problems

Let us assume that the optimal solution to the relaxed RMP had been found and that $\boldsymbol{\pi}^+$, $\boldsymbol{\pi}^-$ $\boldsymbol{\pi}^t$ and $\boldsymbol{\pi}^{t'}$ are the vectors of optimal dual values for constraints (2), (3), (4) and (5) respectively. These vectors are passed to the slave pricing problems in order to produce more routes that will be useful to reduce the value of the objective (1). 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 (ESPPRC) and when there are multiple terminals, a pricing problem may be solved for each terminal in each pricing step. In our application we solve exactly the MILP formulation of the elementary pricing problems with a branch-and-cut solver. What follows is the formulation to the pricing problem for generating pick-up routes:

Minimize

$$\left(CV - \sum_{i \in I^+} \pi_i Y_i - \sum_{r \in R^T} \sum_{t \in T} b_{tr}^{\ start} \sum_{i \in I^+} \pi_{ri}^{t} x_t Y_i - \sum_{r \in R^T} \sum_{t \in T} b_{tr}^{\ start} \sum_{i \in I^+} \pi_{ri}^{t'} x_t T_i^+ \right) \qquad (7)$$

subject to:

$$\sum_{t \in T} x_t = 1 \qquad (8)$$

$$x_t = 1 \qquad t = \text{selected terminal} \qquad (9)$$

$$D_i \geq \sum_{t \in T} x_t d_{ti} \qquad (10)$$

$$\forall i \in I^+$$

$$\begin{Bmatrix} D_j \geq D_i + d_{ij} - M_D (1 - S_{ij}) - M_D (2 - Y_i - Y_j) \\ D_i \geq D_j + d_{ij} - M_D S_{ij} - M_D (2 - Y_i - Y_j) \end{Bmatrix} \qquad \begin{matrix} (11.a) \\ (11.b) \end{matrix}$$

$$\forall (i,j) \in I^+ : i < j$$

$$CV \geq cf_v + D_i + \sum_{t \in T} x_t d_{ti} - M_C (1 - Y_i) \tag{12}$$

$$\forall i \in I^+$$

$$T_i \geq \sum_{t \in T} t_t t_{ti} \tag{13}$$

$$\forall i \in I^+$$

$$\begin{Bmatrix} T_j \geq T_i + st_i + t_{ij} - M_T (1 - S_{ij}) - M_T (2 - Y_i - Y_j) & (14.a) \\ T_i \geq T_j + st_j + t_{ji} - M_T S_{ij} - M_T (2 - Y_i - Y_j) & (14.b) \end{Bmatrix}$$

$$\forall (i,j) \in I^+ : i < j$$

$$TV \geq T_i + \sum_{t \in T} x_t t_{ti} - M_T (1 - Y_i) \tag{15}$$

$$\forall i \in I^+$$

$$t_i^{\min} \leq T_i \leq t_i^{\max} \tag{16}$$

$$\forall i \in I^+$$

$$T_i^+ \geq TV + \sum_{i \in I^+} Y_i st_i - M_T (1 - Y_i) \tag{17}$$

$$\forall i \in I^+$$

$$\sum_{i \in I^+} Y_i l_i \leq q \tag{18}$$

The objective function (7) is the cost CV of the generated route minus the prices π_i collected on the visited pick-up sites; minus the prices π_{ri}^{t} related to the inbound load-flow and minus prices $\pi_{ri}^{t'}$ related to unload time on the selected terminal. The parameter a_{ir} of the master problem becomes the decision variable Y_i of the pricing one. Also the parameter (a_{ir} $t_r{}^+{}_{end}$) of the master problem becomes the continuous variable T_i^+ in the pricing problem. The binary parameter x_t indicates the start/end terminal of the designed tour in eqs. (8)-(9). The constraint (10) set the minimum distance to reach the site $i \in I^+$ as the distance of going directly from the terminal to the location i. The constraints (11) and (12) compute the distances travelled to reach the visited sites $i \in I^+$ and the total cost of the generated route respectively. So, eqs. (11) 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 j ($S_{ij} = 1$), according constraints (11.a), the travelled 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 (11.b) becomes active. If one or both sites are not allocated to the tour, the eqs. (11.a)-(11.b) become redundant. M_D is an upper bound for variables D_i. The eq. (12) computes the route-cost CV by the addition of the fixed vehicle utilization cost cf_v to the travelled-distance-cost up to the terminal to which the vehicle must return. M_C is an upper bound for the variable CV. The timing constraints stated by eqs. (13) to (15) are similar to constraints (10) to (12) 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. (16) 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}]. The eq. (17) adds to the tour time-length a term related to the unload activities on the selected terminal to define the end unload-time for each cargo request. This eq. defines the availability time on the terminal of cargo picked-up from site $i \in I^+$. This time must be coordinated with the sum of the transfer time and the load time for the final delivery. This is done via duals of constraints (5) that modify the unload time of the pick-up tour and the load time of the delivery tour, just in case the request is not fulfilled by a mixed trip. The eq. (18) is a capacity constraint for the vehicle travelling the designed pickup tour.

The objective of the slave problem for generating delivery tours is to find a route r minimizing the quantity stated by the objective function (19).

$$Minimize \qquad (19)$$

$$\left(CV - \sum_{i\in I^-} \pi_i Y_i - \sum_{r\in R^T}\sum_{t\in T} b_{tr}{}^{end} \sum_{i\in I^-} \pi_{ri}^{t} x_t Y_i - \sum_{r\in R^T}\sum_{t\in T} b_{tr}{}^{end} \sum_{i\in I^-} \pi_{ri}^{t'} x_t T_i^- \right)$$

subject to constraints that are similar to constraints (9) to (18) but which are used to design delivery routes. So, we change I^+ by I^- in the domain of the constraints (9)-(18) except eqs. (13) and (17) because eq. (17) is replaced by eq. (20) and eq. (13) replaced by eq. (21):

$$\begin{Bmatrix} T_i^- \geq \sum_{i\in I^-} Y_i st_i - M_T(1-Y_i) \\ T_i \geq T_i^- + t_{ti} \end{Bmatrix} \qquad \forall i \in I^- \qquad (20)$$

$$TV \leq t^{\max} \qquad (21)$$

The parameter $t^{\max}$ indicates the end-time for all kind of activities and the load time T_i^- becomes a problem variable coordinated with T_i^+ by the duals of master constraint (5).

The objective of the slave problem for generating pick-up and delivery tours is to find a route r minimizing the quantity stated by the objective function (20).

$$Minimize\left(CV - \sum_{i\in I^+\cup I^-} \pi_i Y_i \right) \qquad (22)$$

Constraints similar to eqs. (9) to (18) but refereed now to the set $\{I^+ \cup I^-\}$ of pick-up and delivery sites must be considered. Eq. (21) must be also included in this slave problem.

2.3 Branching strategy

The linear relaxation of the RMP may not be integer and applying a standard branch-and-bound procedure to this problem with a given pool of columns may not yield an optimal solution. Also a column pricing favorably may exist but it may not be present in the RMP. To find the optimal solution, columns must be generated after branching. So, according to [4] if the master problem returns a solution that is fractional in the number of used tours k, we branch on this number by creating two child nodes equivalent to the current subspace but with the addition of $\Sigma_r X_r \geq ceil(k)$ and $\Sigma_r X_r \leq floor(k)$ 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. After fixing the number of vehicles, we start to branch according to the Ryan and Foster [5] benching strategy. 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 integrated both branching rules in a hierarchical way. The branching procedure 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. Best first search was the node selection strategy.

2.4 Implementation

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 [5, 6] and were integrated in this work. Minor branching and assembling modifications aimed at replacing the NLP of the [4] MINLP algorithm by the CG [6] procedure and aimed at forbidding the branching combination $Y_i = 0$ for all $i \in I^+ \cup I^-$ were also introduced. Some standard speeding tricks [7] as 'early-termination' and 'time windows reduction' were also implemented. The algorithm uses the CPLEX 11 as the MILP sub-algorithm for generating columns and for computing upper and lower bounds. It was tuned to generate a several columns per master-slave iteration.

Table 1: Settings options of the branch-and-price algorithm

Option	
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
Early termination option	Yes
Multiple columns generated per iteration	Yes
Time-windows reduction	Yes
Maximum number of iterations per branch and price node	100
Maximum number of branch-and-price inspected nodes	100 (root)/ 5 (no-root)
Master problem	Partitioning
Columns pool	Up to 10000

3 A case study

We illustrate the use of the solution procedure on a case study with real data. A transportation company from Santa Fe provides distribution services of non-perishable products to several industrial and service companies in the urban Santa Fe area and surroundings. The operation 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. 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 un-constraining. Service times at pick-up/delivery stops are considered approximately constant, st_i = 20 minutes, and the average urban-travel speed is assumed to be 20 *km/h*. The case study uses data from a working day and involves the fulfillment of 44 transportation requests within the day. We estimated the distance (in *km*) between clients locations and between these locations and both depots by using the Manhattan distance formula jointly with the clients locations on the city map. The whole dataset can be found in [3]. Usually the company performs pickup activities during morning and delivery during afternoon to allow some consolidation work between both stages and to avoid cargo warehousing at night. Time windows usually are not considered and sometimes they can be assigned just to a few clients. A fixed van utilization cost cf_v = $ 200 and a unit distance cost $10/km are here considered. Transfer trips "*Central depot – S. Tomé"* and "*S. Tomé - Central depot*" include transportation and workload costs on both depots and have an associated cost $cf_{long\text{-}haul}$ = $1700/day. Cargo transshipment costs on each depot are cf = $400/day. This case study was solved in [3] considering a rigid time-delimitations between the pick-up, transfer and delivery stages. Some vans were allowed to perform pick-up and delivery tours on long trips starting in the morning and ending at the night. Here, we drop hard time-constraints applied to slave pickup problems and to slave delivery problems and introduce in their objective functions the terms related to duals of the coordinating constraint (5), according to the methodology above presented. Afterwards, we applied the solution algorithm above developed to that case study and generated the solution to be next detailed. The algorithm ran in a 2-core, 2.5 GHz, 6 GB RAM notebook and the mechanism settings used to solve the problems are summarized in Table 1. The solution was obtained in 3088 s (integrality gap = 7.67%) and involves 8 pickup tours, 7 delivery tours, 3 mixed tours and 2 transfer-trips. It implied a total cost of $ 17382. That means, we saved $ 238 with respect to the solution reported in [3]. The solution is summarized in Tables 2 to 5.

Table 2: Pick-up tours

Tour	***Trajectory***	***Tour cost ($)***	***Tour time (')***	***Load (m^3)***
1	C-n11-C	276	55	1.2
2	C-n47-n36-n46-n45-C	707	201	6.7
3	ST-n16-n39-n14-ST	727	213	7.5
4	C-n19-n7-n12-n24-n33-C	967	287	6.0
5	C-n31-n50-n25-n13-n17-C	1027	285	7.2
6	ST-n15-n43-n38-n44-ST	443	149	5.5
7	C-n32-n20-n35-n9-n10-C	653	211	7.3
8	C-n49-n8-n48-C	582	161	7.5

C: Central depot; ST: Secondary S.Tomé depot; Time t = 0' correspond to 8:00 AM

Table 3: Requests transshipped

Depot	*Requests transshipped*	*Load transshipped (m^3)*
C	n7-n19-n12-n33-n31-n50-n13-n17-n32-n20-n35-n9-n10-n49-n48	28.7

Table 4: Requests transferred between both depots

Trip	*Requests transferred*	*Cargo transferred (m^3)*
C → ST	n1-n36-n46-n45-n7-n24-n25-n8	7.2
ST → C	n16-n39-n14-n15-n43-n38-n44	13.0

Table 5: Delivery tours

Tour	*Start time (')*	*Trajectory*	*Tour cost ($)*	*Tour time (')*	*Load (m^3)*
1	375	ST-n57-n74-n75-n67-n58-n95-n96-n61-n86-ST	949	704	7.4
2	488	C-n60-n97-n99-C	662	679	7.5
3	526	C-n94-n89-n88-C	562	724	7.5
4	547	C-n62-n64-n63-C	510	669	6.5
5	513	C-n66-n65-n81-C	787	690	6.0
6	497	C-n70-n69-n82-n83-C	630	663	6.7
7	500	C-n93-n59-n98-n100-n85-C	688	737	7.3

Table 6: Pick-up and delivery tours

Tour	*Trajectory*	*Tour cost ($)*	*Tour time (')*	*Load(m^3)*
1	C-n18^{+}-n27^{+}-n28^{+}-n29^{+}-n34^{+}-n28^{-}-n29^{-}-n27^{-}-n18^{-}-n24^{-}-C	1171	713	72
2	ST-n42^{+}-n23^{+}-n22^{+}-n41^{+}-n37^{+}-n73^{-}-n37^{-}-n41^{-}-n42^{-}-n22^{-}-ST	983	662	68
3	C-n30^{+}-n40^{+}-n21^{+}-n26^{+}-n26^{-}-n30^{-}-n21^{-}-n40^{-}-C	1037	690	62

+Pickup location - Delivery location

4 Conclusions

We developed a truncated branch-and-price solution algorithm to efficiently design a transportation agenda for a LTL-like practical problem involving the fulfillment of a list of transportation requests in an urban area and surroundings by choosing between three different delivery options: direct delivery by the same vehicle, a delivery via transshipment on a terminal or a three-stages delivery option which includes a pick-up step, a long-haul route between two terminals and the final delivery. The problem was first modeled as a set partitioning problem with additional transfer and unloading/transfer/loading time-coordinating constraints. The model was later embedded into an incomplete branch-and-price solution-mechanism. The mechanism reorders the transfer and time-coordination constraints to express them as covering constraints to add to the partitioning constraints for pick-up and delivery locations.

The pricing problems were formulated as integer-linear programs and solved by a branch-and-cut solver trying to obtain a maximum number of elementary columns per master-slave iteration. The work was built over a previous one [3] that considers that pick-up activities must end before a stated timeline t^{max+} and delivery activities must start after a timeline t^{min-} ($>t^{max+}$). The interval between both timelines is devoted to transshipment and transfer activities. In the present work we drop these hard constraints from the associated slave pick-up problem and slave delivery problem. Since pick-up and delivery tours must be now coordinated, the dropping of these constraints from slave problems means the introduction of an additional coordinating constraint in the master problem. This constraint, in turn, passes information to slave subproblems via duals $\pi_{ri}^{t'}$ that are useful to adjust the end-time of pickup routes and the start time of delivery tours. Some standard options were also taken: branching on the number of tours was selected as a higher level branching-rule to explore a finite branch-and-price tree. After fixing the number of vehicles, the algorithm starts to branch according the Ryan and Foster rule. The use of the mechanism was illustrated by solving a case study previously solved in a framework that strictly time-delimit the pickup, transfer and delivery phases for trips others than the mixed one. A small cost saving was obtained with respect to this older framework. The procedure proposed in this work was aimed at eliminating these rigid delimitations. Further numerical examples should be solved to evaluate the robustness of the procedure.

References

1. Dondo R., Mendez C., Cerdá J., The Multi-Echelon Vehicle Routing Problem with Cross-Docking in Supply Chain Management". *Computers and Chemical Engineering*, 35 (12), 3002-3024 (2011).

2. Erera A., Hewitt M., Savelsbergh M., Zhang Y., Creating schedules and computing operating costs for LTL load plans. *Computers and Operations Research*, 40, 691–702 (2013).

3. Dondo R. and Mendez C. A Branch-and-Price Approach To Manage Cargo Consolidation and Distribution in Supply Chains. *Industrial & Engineering Chemistry Research.* 53, 17226-17239 (2014).

4. 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, 269−280 (1981).

4. Desrochers M., Desrosiers J. and M. Solomon.. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 12 (2), 342-354 (1992).

5. Kalvelagen E. Columns generation with GAMS Downloaded from http://amsterdamoptimization.com/pdf/colgen.pdf. (2003).

6. Kalvelagen E. Some MINLP Solution algorithms Downloaded from http://www.amsterdamoptimization.com/pdf/minlp.pdf. (2003).

7. Desaulniers G., Desrosiers J.. Solomon M. Accelerating Strategies in Column Generation Methods for Vehicle Routing and Crew Scheduling Problems. *Operations Research/Computer Science Interfaces Series* 15, 309-324 (2002).