

A branch-and-price method for the Vehicle Routing Problem with Cross-Docking and Time Windows

Rodolfo Dondo¹

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

Abstract. One important factor in supply chain management is to efficiently control the supply chain flows. Due to its importance, many companies are trying to develop efficient methods to increase customer satisfaction and reduce costs. Cross-docking is considered a good method to reduce inventory and improve responsiveness. The Vehicle Routing Problem with Cross-Docking and Time Windows (VRP-CD-TW) consists on designing the minimum-cost set of routes to serve a given set of transportation requests while respecting constraints on vehicles capacity, customer time windows and using transfers on a cross-docking base. Each customer must be visited just once and mixed tours comprising pick-up and delivery stops are not allowed. For a given vehicle, the designed pick-up tour must precede its delivery tour. In this work, we model the VRP-CD-TW assuming that all feasible orders are known in advance. We present a new mixed integer program to model the VRP-CD-TW and reformulate it via Dantzig-Wolfe decomposition to later develop a column generation procedure. The proposed branch-and-price algorithm shows encouraging results on solving some Solomon-based instances.

Keywords: Supply-chain management, cross-docking, vehicle routing, columns generation.

1 Introduction

Cross-docking has emerged as an important and efficient goods transportation strategy. As a variation of the well-known vehicle routing problem (VRP), the VRP with Cross-docking (VRPCD) arises in a number of logistic planning contexts. At cross-docking terminals incoming deliveries of inbound trucks are unloaded, sorted, moved across the dock and finally loaded onto outbound trucks, which immediately leave the terminal towards their next destination in the distribution chain. The cross-dock is a consolidation point in a distribution network, where multiple smaller shipments can be merged to full truck loads in order to obtain savings in transportation [1]. The objective of the VRP-CD-TW consists of designing the minimum-cost set of routes to serve a given set of transportation requests while fulfilling constraints on vehicles capacity and customer time windows and using goods transfers on a cross-docking base. As cross-docking is a comparatively new logistic strategy, there is not yet a massive body of literature on the subject. In fact, no

research on the short-term truck-scheduling-problem was published before 2005. Lee et al. [2] considered cross-docking from an operational viewpoint in order to find the optimal vehicle routing schedule. Thus, an integrated model considering both cross-docking and vehicle routing scheduling was presented. Since the problem is NP-hard, a heuristic algorithm based on a taboo search algorithm was proposed. One of the objectives for cross docking systems is how well the trucks can be scheduled at the dock and how the items in inbound trucks can be allocated to the outbound trucks to optimize on some measure of system performance [3]. The authors researched on how to find the best truck docking or scheduling sequence for both inbound and outbound trucks to minimize total operation time when a storage buffer to hold items temporarily is located at the shipping dock. The product assignment to trucks and the docking sequences of the inbound and outbound trucks are all determined simultaneously. Wen et al. [4] presented a very detailed mixed formulation for the problem and a taboo search heuristic embedded within an adaptive memory procedure. The procedure was tested on data sets unreported in the paper that were provided by the Danish Transvision consultancy involving up to 200 pair of nodes. Liao et al. [5] presented a new taboo search (TS) algorithm that was proposed to obtain a good feasible solution for the problem. Through extensive computational experiments, they claim that the proposed TS algorithm can achieve better performance than a previous TS by Lee et al. algorithm while using much less computation time. Boloori Arabani et al [6] studied some meta-heuristics to find the best sequence of inbound and outbound trucks, so that the objective, minimizing the total operation time or makespan, can be satisfied. In this work, we study the VRP-CD-TW and propose a branch-and-price algorithm for solving such a problem. In section 2 we formulate the problem. In section 3 we present and describe the branch-and-price algorithm devised to solve the CRP-CD-TW. Computational results are presented and discussed in section 5 and the conclusions are outlined in section 4.

2 Problem Formulation

Let $G[I; A]$ a directed graph involving the locations set $I = \{w \cup I^+ \cup I\}$ and the net $A = \{a_{ii'} : i, i' \in I \text{ and } i \neq i'\}$. I contains the the cross-dock-base w , the pick-up nodes $I^+ = \{i_1, \dots, i_n\}$ and the delivery nodes $I = \{i'_1, \dots, i'_n\}$. A request $r = \{i, i'\}$ of a request list $R = \{r_1, \dots, r_r\}$ consists of a demand of a transportation service from the origin-node(s) $i \in I^+$ to the destination node(s) $i' \in I$ for certain load l_i . Each arc $a_{ii'} \in A$ have an associated non-negative cost c_{ij} and an associated non-negative travel-time t_{ij} . The service time on each node $i \in I$ is defined by the parameter st_i . Requests must be fulfilled through a homogeneous vehicles fleet $V = \{v_1, v_2, \dots, v_m\}$. The solution consists of a finite set of sequences of arcs, called *routes*, for some vehicles $v \in V$ such that: (i) Each vehicle can perform pick-up and/or delivery tours but each individual tour contains either pick-up or delivery locations. No "mixed" routes with both type of locations are allowed. (ii) For each vehicle, the pick-up tour must precede its designed delivery route. (iii) Each vehicle starts and ends the pick-up and/or delivery routes at the cross-dock base w . (iv) Each pick-up/delivery site $i \in I^+ \cup I$ is assigned

to exactly one route. (v) The actual load carried by a vehicle v must never exceed its transport capacity q_v . (vi) The service for any pick-up/delivery node $i \in I^+ \cup I^-$ must start within the time-window $[a_i, b_i]$; (vii) The whole duration of the vehicle- v trip, including the pick-up tour, the transfer operations and the delivery tour must be shorter than a maximum routing time t_{v}^{max} . (viii) The problem goal is to minimize the total cost for providing pickup/ delivery service to every node $i \in I^+ \cup I^-$.

3 Column Generation

In this section we introduce a set partitioning formulation of the problem above presented to later provide the formulations of the master problem and the pricing sub-problem.

3.1 The Master Problem

The definition of the master problem requires the following notation: c_r^+ is the cost of the pick-up route r^+ and a_{ir} denotes a binary parameter equal to 1 if the route r^+ picks-up the cargo of the request $r \in R$. In the same way, c_r^- is the cost of the delivery route r^- while b_{ir} denotes a binary parameter equal to 1 if the route r^- delivers the cargo of the request $r \in R$. After the introduction of the time-coordination constraint (4), the master problem can be formulated as follows:

$$\text{Minimize } \sum_{r \in R} c_r^+ x_r^+ + \sum_{r \in R} c_r^- x_r^- \quad (1)$$

Sujeto to

$$\sum_{r \in R} a_{ir} x_r^+ = 1 \quad \forall i \in I^+ \quad (2)$$

$$\sum_{r \in R} b_{ir} x_r^- = 1 \quad \forall i \in I^- \quad (3)$$

$$\sum_{r \in R} a_{ir} t_{end}^+ x_r^+ \leq \sum_{r \in R} b_{jr} t_{start}^- x_r^- \quad \forall i \in I^+, j \in I^- : (i, j) = r \in R$$

$$x_r = \{0,1\}$$

where x_r^+ and x_r^- are binary variables associated respectively to pick-up routes r^+ and r^- . The dual variables associated to constraints (2) and (3) are both collected in the vector $\pi = [\pi_1, \dots, \pi_i, \dots]$ while $\pi_r = [\pi_{r_1}, \dots, \pi_{r_r}]$ collect the dual values associated to

coordination constraint (4). The objective function (1) selects the set of pick-up and delivery routes that minimizes the total traveling cost. Constraints (2) ensure that all requests $r \in R$ are picked from pick-up sites $i \in I^+$, while constraint (3) ensure that all requests $r \in R$ are delivered to delivery locations $i \in I^-$. Constraint (4) coordinates the pick-up and delivery tours that move a request from its origin to its destination. So, the end of the pick-up tour must precede the time-start of the delivery tour. We remark that constraints (2) and (3) need to be modeled as partitioning constraints in the VRP-CD-TW, unlike common reformulations for routing problems that generally make use of covering constraints. This is due to the fact that a linear combination of active pick-up and delivery routes must lead to a solution on which the pick-up tour starts before the end of the delivery route. As a consequence, a partitioning solution equivalent to the optimal covering solution may be infeasible.

3.2 Pricing sub-problem

In a column generation scheme, given a dual solution of the (restricted) master problem, the pricing sub-problem identifies the route r^* with the minimum reduced cost:

$$c_{r^*} = CV^+ + CV^- - \sum_{i \in I^+} \pi_i^+ Y_i - \sum_{i \in I^-} \pi_i^- Y_i - \sum_{i \in I^+} \sum_{i \in I^- : (i,j)=r \in R} \pi_r YR_r$$

The sub-problem formulation relies on the some integer and continuous variables and can be written as follows

$$\begin{aligned} & \text{Minimize} \\ & CV^+ - \sum_{i \in I^+} \pi_i^+ Y_i + CV^- - \sum_{i \in I^-} \pi_i^- Y_i - \sum_{i \in I^+} \sum_{i \in I^- : (i,j)=r \in R} \pi_r YR_r \end{aligned} \quad (5)$$

Subject to:

$$D_i \geq d_{0i} \quad \forall i \in I^+ \quad (6)$$

$$\left\{ \begin{array}{l} D_j \geq D_i + d_{ij} - M_D(1 - S_{ij}) - M_D(2 - Y_i - Y_j) \\ D_i \geq D_j + d_{ij} - M_D S_{ij} - M_D(2 - Y_i - Y_j) \end{array} \right\} \quad (7.a)$$

$$\forall i, j \in I^+ : i < j \quad (7.b)$$

$$CV \geq D_i + d_{i0} - M_C(1 - Y_i) \quad \forall i \in I^+ \quad (8)$$

$$T_i \geq t_{0i} \quad \forall i \in I^+ \quad (9)$$

$$\left\{ \begin{array}{l} T_j \geq T_i + st_i + t_{ij} - M_T(1 - S_{ij}) - M_T(2 - Y_i - Y_j) \\ T_i \geq T_j + st_j + t_{ji} - M_T S_{ij} - M_T(2 - Y_i - Y_j) \end{array} \right\} \quad (10.a)$$

$$\forall i, j \in I^+ : i < j \quad (10.b)$$

$$TV^+ \geq T_i - M_T(1 - Y_i) \quad \forall i \in I^+ \quad (11)$$

$$\sum_{i \in I^+} l_i Y_i \leq Q \quad (12)$$

$$YR_r \leq Y_i^+ \quad (13.a)$$

$$YR_r \leq Y_j^- \quad (13.b)$$

$$YR_r \geq Y_i^+ + Y_j^- \quad (13.c)$$

$$t_{end}^+ \geq TV^+ + \sum_{i \in I^+} st_i^0 Y_i - \sum_{i \in I^+} \sum_{i \in I^-: (i,j)=r \in R} st_i YR_r \quad (14.a)$$

$$t_{start}^- \geq t_{end}^+ + \sum_{i \in I^-} st_i^0 Y_i - \sum_{i \in I^+} \sum_{i \in I^-: (i,j)=r \in R} st_j YR_r \quad (14.b)$$

$$D_i \geq d_{0i} \quad \forall i \in I^- \quad (15)$$

$$\left\{ \begin{array}{l} D_j \geq D_i + d_{ij} - M_D(1 - S_{ij}) - M_D(2 - Y_i - Y_j) \\ D_i \geq D_j + d_{ij} - M_D S_{ij} - M_D(2 - Y_i - Y_j) \end{array} \right\} \quad (16.a)$$

$$\forall i, j \in I^- : i < j \quad (16.b)$$

$$CV \geq D_i + d_{i0} - M_C(1 - Y_i) \quad \forall i \in I^- \quad (17)$$

$$T_i \geq t_{start}^- + t_{0i} \quad \forall i \in I^- \quad (18)$$

$$\left\{ \begin{array}{l} T_j \geq T_i + st_i + t_{ij} - M_T(1 - S_{ij}) - M_T(2 - Y_i - Y_j) \\ T_i \geq T_j + st_j + t_{ji} - M_T S_{ij} - M_T(2 - Y_i - Y_j) \end{array} \right\} \quad (19.a)$$

$$\forall i, j \in I^- : i < j \quad (19.b)$$

$$TV^- \geq T_i - M_T(1 - Y_i) \quad \forall i \in I^- \quad (20)$$

$$\sum_{i \in I^-} l_i Y_i \leq Q \quad (21)$$

$$TV^- \leq t^{\max} \quad (22)$$

$$a_i \leq T_i \leq b_i \quad \forall i \in I \quad (23)$$

The objective (5) identifies the route with minimum reduced cost to perform by a single vehicle. Prices values are obtained from the dual values for constraints (3), (4) and (5). The cost constraints of the pick-up tour are given by eqs (6), (7) and (8). They compute the distances travelled to reach the visited sites $i \in I^+$ and the total cost of the generated route respectively. So, eqs. (7) fix the accumulated distance up to each visited site. I.e. if nodes i and j are allocated to the generated pick-up 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 (7.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 (7.b) becomes active. If one or both nodes are not allocated to the tour, eqs (7.a)-(7.b)

become redundant. M_D is an upper bound for variables D_i . Eq (8) transforms the total travelled distance into the route-cost CV . M_C is an upper bound for the variable CV . The *timing constraints* stated by eqs. (9), (10) and (11) define visiting-time constraints that are similar to constraints (6), (7) and (9) but apply to the time dimension. M_T is an upper bound for the times T_i spent to reach the nodes $i \in I$. Eq. (12) is the cargo-capacity constraint related to the pick-up tour. Eqs. (13) activate the variables YR_r indicating that the cargo r is picked-up and delivered by the same vehicle. In such a case, the request cargo will remain on the truck and no drop-off/pick-up activities are incurred. Constraint (14.a) states that the time-end of the pick-up phase must be the sum of the time spent completing the pick-up tour plus the time incurred unloading cargos at the cross-dock base. Conversely, eq. (14.b) states that the delivery stage must start after the sum of time t_{end}^+ and the time incurred in loading goods to be later delivered. Constraints (15)-(17) are cost constraints related to the delivery tour while eqs. (18)-(20) define time constraints related to this delivery tour. Eq. (21) apply the vehicle capacity constraint to the delivery tour and eq. (22) forces the service time at any pick-up/delivery site $i \in I$ to start at a time T_i bounded by the time interval $[a_i, b_i]$. Finally, the eq. (23) imposes the upper bound t_v^{max} to the total tour-time.

3.3 Branch and Price Implementation

The Foster and Ryan branching rule is very favourable for branch and price applications and fits easily with the VRP-CD-TW pricing problem since it is equivalent to branch on assignment decisions Y_i , for all $i \in I^+ \cup I^-$. This branching rule is implemented as follows: Routes without any branching constraint are generated in the root node until no more columns can be obtained. After the bounds comparison in this node, a location is chosen based on a usual branching criterion (i.e. best first search) to later generate two sub-problems in the second tree-level. The first sub-problem generates routes that include a second chosen location while the second sub-problem generates a route without it. In the third level another location is introduced rising to $(2^{n-1} - 1) = 3$ the number of nodes of the branch-and-price tree. The worst case propagation involves $(2^{n-1} - 1)$ nodes at the n level of the tree. To prune this tree, the global upper bound (GUB) and the local lower bound (LLB) are computed on each tree-node. The lower bound within a branch-and-price node is found by solving the linear problem defined by eqs. (1)-(4) on the subset of columns that fulfil the frozen assignment decisions of the node. That is easy because linear RMP involving several thousand columns are solved within a small fraction of a second using modern CPLEX codes in standard PCs. The GUB is more difficult to compute because involves the resolution of an integer RMP on all generated columns. Several computation techniques aimed at avoiding the complete problem resolution have been proposed. Nevertheless, the best upper bound can be computed by solving the integer RMP, branching on the column selection variables x_r , because if the columns set remains in moderate sizes (i.e. a few thousand columns) this problem can be solved in a few seconds with state of the art branch-and-cut solvers. If the size of the columns poll grows above this threshold, some columns can be ignored according a filtering criterion. The filtering leads to a slimmed integer problem that provides a

heuristic upper bound that may coincide with the optimal bound. This frequently occurs if the columns selection is correctly carried out.

4 Computational Results

The algorithm was coded and compiled with Gams 26.6.2 and the problems run under a Windows 7 operating system on a 2 GHz 16 GB RAM PC. To the best of our knowledge there are no standard datasets used to evaluate VRP-CD-TW solution algorithms. Some cited instances as those provided by the Danish Transvision Group [4] are not openly available. Consequently, we generated our test bed from the well-known VRPTW Solomon's test-bed [7].

To generate VRP-CD-TW instances, we took the first $n = 10, 20, 30, 40$ and 50 customers of the Solomon's R1-class problems. For each instance, the first half of them are considered pick-up sites while the second half are regarded as delivery locations. The service times at pick-up and delivery nodes are constants kept at the value $st_i = 10$ taken from the Solomon instances. Also, the maximum routing time is kept at the value $t_v^{max} = 230$. In addition, different time-windows within with the pick-up and delivery services must start, are also considered. They are taken from the Solomon instances R101, R105 and R109. As pick-up tours must precede delivery tours, some time-windows will be inconsistent and must be eliminated. In such a case, no time windows must be considered on the delivery locations. The Table 1 presents a summary of the instances solved by the branch-and-price procedure.

Table 1. Summary of the solutions found by the Branch-and-price procedure on the generated.

| Nodes | Time Windows | Integer Solution | Linear Solution | % Gap | Columns | B & B nodes | Total CPU time |
|-------|--------------|------------------|-----------------|-------|---------|-------------|----------------|
| 10 | - | 233.0 | 229.4 | 1.5 | 56 | 1 | 6.0 |
| | 101 | 254.3 | 254.3 | 0.0 | 35 | 1 | 8.5 |
| | 105 | 239.5 | 230.2 | 3.9 | 86 | 1 | 3.4 |
| | 109 | 239.5 | 238.3 | 0.5 | 80 | 1 | 3.7 |
| 20 | - | 446.7 | 427.3 | 4.3 | 839 | | 1020.4 |
| | 101 | 515.7 | 474.0 | 8.0 | 192 | 5 | 42.9 |
| | 105 | 493.5 | 462.0 | 6.4 | 631 | 5 | 119.9 |
| | 109 | 487.7 | 444.1 | 8.9 | 730 | 3 | 151.3 |
| 30 | - | 795.8 | 728.0 | 8.5 | 434 | 1 | 35.6 |
| | 101 | 860.3 | 786.8 | 8.5 | 198 | 3 | 73.7 |
| | 105 | 780.0 | 723.1 | 7.3 | 1459 | 5 | 321.6 |
| | 109 | 757.9 | 697.4 | 7.9 | 464 | 2 | 43.6 |
| 40 | - | 967.3 | 913.7 | 5.2 | 3898 | 5 | 3431.3 |
| | 101 | 1188.7 | 1078.1 | 9.3 | 2583 | 5 | 2221.8 |
| | 105 | 1100.7 | 1003.4 | 8.8 | 3272 | 5 | 2455.2 |
| | 109 | 1025.9 | 945.9 | 7.8 | 3713 | 5 | 2716.8 |
| 50 | - | 1252.2 | 1165.7 | 6.9 | 1297 | 5 | 639.4 |
| | 101 | 1564.8 | 1404.0 | 10.3 | 4410 | 5 | 2786.9 |
| | 105 | 1440.2 | 1289.7 | 10.4 | 1548 | 5 | 668.9 |
| | 109 | 1356.3 | 1234.0 | 9.0 | 1320 | 5 | 828.0 |

5 Conclusions

In this work we presented a branch-and-price algorithm for solving the VRP-CD-TW. The branch-and-price algorithm has been implemented by adapting state-of-the-art techniques to the specific structure and properties of the problem. Computational results have shown that our algorithm is quite efficient in solving moderate-size instances. By analyzing the computational results, we can conclude that the problem is quite complex and therefore, hard to solve. Nevertheless, we managed to solve instances with up to 50 customers in reasonable CPU times. Although more efficient solution techniques could be explored, we consider these results satisfactory and a good starting point for investigating more sophisticated approaches in the future.

References

1. Boysen N. and M. Fliedner. Cross-dock scheduling: Classification, literature review and Research Agenda. *Omega* 38, 413-422 (2010).
2. Lee Y., Jung W.J., and K.M. Lee. Vehicle Routing Scheduling for Cross-Docking in the Supply Chain. *Comp. and Ind. Eng.*, 51, 247-256 (2006).
3. Yu W. and P. Egbelu. Scheduling of Inbound and Outbound Trucks in Cross-Docking Systems with Temporary Storage. *Eur. J. of Op. Res.* 184, 377-396 (2008).
4. Wen, M., Larsen, J., Clausen, J., Cordeau, J.-F., & Laporte, G. Vehicle routing with cross-docking. *Journal of the Operational Research Society*, 60,1708–1718 (2009).
5. Liao, Ch.-J., Lin, Y., & Shih, S. C. Vehicle routing with cross-docking in the supply chain. *Expert Systems with Applications*, 37, 6868–6873 (2010).
6. Bolori Arabani A., Fatemi Ghomi S. and M. Zandieh. Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Exp. Sys. with Appl.* 38, 1964–1979 (2011).
7. Solomon, M. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 32, 254–265 (1987).