Computers and Chemical Engineering 48 (2013) 293–311

Contents lists available at SciVerse ScienceDirect

Computers and Chemical Engineering

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

A sweep-heuristic based formulation for the vehicle routing problem with cross-docking

Rodolfo Dondo, Jaime Cerdá *

INTEC (UNL – CONICET) Güemes 3450 – 3000, Santa Fe, Argentina

ARTICLE INFO

Article history:
Received 3 January 2012
Received in revised form 27 July 2012
Accepted 25 September 2012
Available online xxx

Keywords:
Cross-docking
Supply chain management
Vehicle routing
Scheduling

ABSTRACT

Cross-docking is a warehousing strategy in logistics used by process industries making products with high proportions of distribution costs. It is described as the process of moving goods from suppliers to customers through a cross-dock terminal without a long-term storage in this facility. The vehicle routing problem with cross-docking (VRPCD) consists of fulfilling a set of transportation requests using a fleet of homogeneous vehicles to sequentially accomplish the pickup and delivery tasks. Between those operations, there is a consolidation process of incoming shipments at the cross-dock. This work introduces a monolithic formulation for the VRPCD that determines pickup/delivery routes and schedules simultaneously with the truck scheduling at the terminal. To derive a more efficient formulation, a constraint set mimicking the widely known sweep algorithm was incorporated into the rigorous model. The resulting model based on the sweep heuristic can find near-optimal solutions to large problems at very acceptable CPU times.

1. Introduction

Cross-docking is a logistics strategy that seeks to reduce inventory and improve customer satisfaction. It is described as the process of moving goods from suppliers to customer locations through a cross-dock terminal without storing products for very long in this facility. After scanning and sorting goods according to their destinations, incoming shipments move across the cross-dock to the shipping doors. There, they are loaded onto outbound trucks that immediately start their delivery routes. Success stories on cross-docking have been reported by several industries with significant proportions of distribution costs like food and beverage producers, pharmaceutical companies, automobile manufacturers and retail chains. A real case study from the food industry was recently presented by Boysen (2010). A particular feature of frozen foods and other refrigerated products, e.g. pharmaceuticals, is that the cooling chain must be kept intact. Once a shipment is unloaded at the intermediate facility, it must be instantaneously loaded into a cooled outbound trailer. No intermediate storage inside the uncooled terminal is allowed. Cross docking systems also work well for perishable products that need to reach the marketplace faster to preserve quality and freshness.

The vehicle routing problem with cross-docking (VRPCD) involves the fulfilment of a set of known customer orders or requests, each one characterized by the cargo size, the pickup point and the place where it has to be delivered. Time windows for starting the service at pickup and receiving nodes can also be specified. The orders are not driven directly from the pickup place to the delivery location, but they have to go first to a cross-dock terminal. At this facility, there is a fleet of vehicles that sequentially accomplishes the required pickup and delivery tasks. In other words, a vehicle starting from the cross-dock first collects several requests at their pickup points, drives back to the depot, and unloads some but not necessarily all orders. Some loads may remain in the truck if the same vehicle will transport them to their destinations. Then, the truck moves to the assigned shipping door, loads some additional requests and goes out again to serve the delivery locations. After completing their tours, delivery vehicles return to the cross-dock facility. At the end of the scheduling horizon, the overall load that has unloaded at the cross-dock should be equal to that delivered from the cross-dock to the receiving nodes. Split deliveries are not allowed and each pickup/receipt location must be visited by a single vehicle only once. Besides, the overall load transported by a vehicle must never exceed its capacity. The problem objective is to determine the best pickup and delivery routes as well as the arrival times of pickup/delivery vehicles at the cross-dock so that all nodes are visited within their time windows at minimum total transportation cost, including variable and fixed costs. Moreover, all delivery tasks should be completed before the end of the planning horizon. An alternative goal is the minimization of the overall makespan (MK) for the whole planning task, where MK is defined by the outbound vehicle last arriving at the cross-dock.

* Corresponding author.
E-mail address: jcerda@intec.unl.edu.ar (J. Cerdá).

0098-1354/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compchemeng.2012.09.016

Nomenclature

Sets

N	events at the cross-dock terminal
R	customer requests
V	vehicles

Parameters

$d^P_{r,r1}$	travel distance between the pickup nodes of requests (r,$r1$)
$d^D_{r,r1}$	travel distance between the delivery nodes of requests (r,$r1$)
$d^P_{w,r}$	travel distance between the cross-dock and the pickup node of request r
$d^D_{w,r}$	travel distance between the cross-dock and the delivery node of request r
ft^P_r	fixed portion of the service time at the pickup node of request r
ft^D_r	fixed portion of the service time at the delivery node of request r
ft^P_w	fixed portion of the length of unloading operations at the cross-dock
ft^D_w	fixed part of the length of loading operations at the cross-dock
lr_r	loading rate at the pickup node of request r
lr_w	loading rate at the cross-dock
M_C/M_T	big-M for routing cost and routing time constraints
q_r	size of request r
Q_v	maximum capacity of vehicle v
sp_v	average speed of vehicle v
uc_v	transportation cost per unit travel distance for vehicle v
ur_r	unloading rate at the delivery node of request r
ur_w	unloading rate at the cross-dock
θ^P_r	angular coordinate of the pickup node of request r
θ^D_r	angular coordinate of the delivery node of request r

Binary variables

$XP_{r,r1}$	sequencing nodes on the same pickup route
$XD_{r,r1}$	sequencing nodes on the same delivery route
U^P_v	denoting the usage of pickup vehicle v
U^D_v	denoting the usage of delivery vehicle v
WP_{nv}	allocating event n to pickup vehicle v
WD_{nv}	allocating outbound vehicle v to event n
YP_{rv}	assigning pickup request r to vehicle v
YD_{rv}	assigning delivery request r to vehicle v
ξ^P_r	denoting that the pickup node of request r belongs to the last sector
ξ^D_r	denoting that the delivery node of request r belongs to the last sector

Continuous variables

AT^P_v	arrival time of pickup vehicle v at the cross-dock
AT^D_v	arrival time of delivery vehicle v at the cross-dock
C^P_r	arrival time of inbound vehicle v at the pickup node of request r
C^D_r	arrival time of outbound vehicle v at the delivery node of request r
I_{rnv}	amount of request r unloaded at the cross-dock from vehicle v up to event n
IT_{rn}	amount of request r unloaded at the cross-dock up to event n
OC^P_v	total traveling cost for the pickup route v
OC^D_v	overall traveling cost for the delivery route v
DV^P_r	angular correction for request r moved to an adjacent pickup tour
DV^D_r	angular correction for request r moved to an adjacent delivery tour
RT^P_v	release-time of pickup vehicle v from its pickup duties
T^P_r	vehicle arrival time at the pickup node of request r
T^D_r	vehicle arrival time at the delivery node of request r
TE_n	time of event n at the cross-dock
YR_{rv}	denoting that request r has been fully assigned to vehicle v
ϕ^P_v	lower angular limit of the pickup sector v
ϕ^D_v	lower angular limit of the delivery sector v
$\Delta\phi^P_v$	angular width of the pickup sector v
$\Delta\phi^D_v$	angular width of the delivery sector v
$\Delta\theta$	upper bound on the angular correction for requests moved to nearby sectors

To effectively apply the cross-docking strategy, pickup and delivery tasks and the consolidation process at the cross-dock must be simultaneously considered. To avoid delays on the start of the delivery tasks, some solution techniques for the VRPCD problem assume the simultaneous arrival of pickup vehicles at the cross-dock to develop the delivery routes. In practice, such a condition can only be fairly approximated. If all vehicles do not arrive at the terminal simultaneously, some vehicles have to wait. If they arrive at the same time, the consolidation process at the cross-dock can be rapidly accomplished and goods can be moved to their destinations without delay and storage. As a result, the average inventory level at the cross-dock and the lead-time for delivery both diminish. Short lead times for delivery must be guaranteed because little stock can be held at the cross-dock facility. Another usual assumption of VRPCD methods is that all vehicles have an identical capacity, i.e. homogeneous vehicles.

The VRPCD problem includes three interrelated logistics subproblems: (a) the pickup vehicle routing and scheduling; (b) the short-term truck scheduling at the cross-dock, and (c) the delivery vehicle routing and scheduling. The three subproblems must be simultaneously tackled in order to improve the material flow from suppliers to retailers. The consolidation of goods at the cross-dock terminal considered by subproblem-(b) may slow down the distribution operations and produce a significant amount of double handling. An efficient transshipment process requires harmonizing the timing and the requests transported by inbound and outbound vehicles so that the average inventory inside the terminal is kept low and on-time deliveries are ensured. Generally, the cross-dock has multiple receiving and shipping doors but still lower than the number of vehicles. Then, trucks are queued and waiting for the assignment of an empty door to start loading or unloading operations at the cross-dock. Therefore, the truck scheduling subproblem should decide on (i) the assignment of vehicles to doors at the cross-dock, (ii) the grouping of incoming shipments based on their destinations and the subsequent allocation to outbound vehicles, and (iii) the sequence of pickup or delivery trucks waiting for service at the same dock door. These doors are equipped with enough equipments (e.g. hand stackers or fork lifts) and workers to process trucks once at a time. Between the unloading of pickup vehicles and the loading of delivery trucks, there is a time lag for material handling inside the terminal. This time lag surely depends on the set of loads collected by each pickup vehicle and its arrival time at the cross-dock. The truck scheduling problem assumes that the depot is given and the number of dock doors and their placement along the perimeter of the depot are known. Moreover, the types and quantities of products transported by inbound and outbound vehicles are also problem data. Therefore, the truck scheduling problem should decide on how goods are exchanged between inbound and outbound trailers. Besides, the distance between any pair of doors is given and the time lag for material handling between any pair of doors can be fairly estimated.

Planning inbound and outbound vehicle routing and scheduling simultaneously with the truck scheduling at the cross-dock through a monolithic optimization model seems to be a very interesting challenge. Most previous contributions have focused on the isolated truck scheduling. However, vehicle routing adds considerable degrees of freedom by varying arrival and departure times of trucks, and accounting for them it may produce a significant improvement on the overall planning task (Lee et al., 2006). This work presents a monolithic approach for the VRPCD problem based on a mixed-integer linear programming (MILP) formulation. The model is capable of selecting inbound and outbound vehicle routes and schedules and precisely considering the time interval during which some orders should stay in the temporary storage of the cross-dock. There, they will be waiting for the delivery truck or for the arrival of the remaining goods to be loaded on the assigned vehicle. The proposed model will assume that the cross-dock facility has an unlimited number of doors so that every arriving truck can immediately start loading or unloading operations. Then, such a contribution to the time lag at the cross-dock will be neglected. The problem target is to minimize either the total pickup and delivery routing costs or the overall makespan of the whole planning task. To derive a much more efficient MILP model by easing the process of allocating vehicles to nodes, a set of equations mimicking the sweep-heuristic algorithm of Gillett and Miller (1974) has been embedded into the problem formulation. The resulting MILP sweep-based formulation (SBF) is able to find near-optimal solutions to large problems at low CPU times.

2. Previous contributions

Considerable research on cross-docking has been made in the past. However, most of the papers have focused on strategic issues such as the layout of the cross-dock including the number of dock doors and the shape of the terminal building (Bartholdi & Gue, 2004; Ratliff, Vate & Zhang, 1990), and the best locations for such facilities (Gumus & Bookbinder, 2004; Jayaraman & Ross, 2003; Ross & Jayaraman, 2008; Sung & Song, 2003). Another problem with many contributions at the tactical and operational levels is truck scheduling. A thorough review can be found in Boysen and Fliedner (2010). In truck scheduling, the number and placement of dock doors along the perimeter of a cross-dock terminal are known. Moreover, the size and composition of the cargo transported by inbound/outbound trucks are also problem data rather than problem variables. The major goal is to transfer products from incoming trucks to outgoing trucks on the cross-dock at the least operational time, i.e. the makespan. A temporary storage buffer in front of the shipping dock is assumed to be available. The allocation of vehicles to receiving/shipping docks and their arrival/departure times, the sequencing of inbound/outbound trucks waiting for service at the same dock door and the exchange of products between incoming and outgoing vehicles are the major decision variables. However, the vehicle routing problem is not considered. Goods can be directly conveyed from inbound to outbound trucks without being kept in the temporary storage, or they can be stored in the temporary storage and later loaded onto outbound trucks. Minimizing the number of units passing through the temporary storage is one of the operational targets. Mixed integer programming models for truck scheduling problems with small or medium sizes, and meta-heuristic approaches for large-size case studies have been proposed (Boloori Arabani, Fatemi Ghomi, & Zandieh, 2011; Boysen, 2010; Boysen, Fliedner, & Scholl, 2010; Li, Low, Shakeri, & Lim, 2009; Tsui & Chang, 1992; Yu & Egbelu, 2008). Different meta-heuristics like genetic (GA), tabu search (TS), particle swarm optimization (PSO), ant colony optimization (ACO) and differential evolution (DE) algorithms has been tested by solving large size problems. The related multi-cross-dock transshipment service problem with time windows involves a number of suppliers, cross-docks and customers (Chen, Guo, Lim, & Rodrigues, 2006; Lim, Miao, Rodrigues, & Xu, 2005; Miao, Yang, Fu, & Xu, 2012). Known supplies and demands are taken as deliveries from suppliers and pickups by customer to occur at cross-dock facilities within specific time windows. In other words, the flows from suppliers to customers via the cross-docks are constrained by fixed transportation schedules and limited inventory capacities that cannot be exceeded. The objective is to find a minimum cost distribution plan that meets all demands at minimum transportation and inventory-handling costs. As supplies and demands are not necessarily equal, the level of inventory may change over the planning horizon. Major decision variables are the assignment of suppliers/customers to cross-docks and the amount of products available on every cross-dock at any time.

On the other hand, few papers have dealt with the integrated vehicle routing problem with cross-docking (VRPCD). Lee, Jung and Lee (2006) were the first authors to study the VRPCD problem. They developed an MILP integrated model that considers cross-docking operations and vehicle routing and scheduling, assuming that all vehicles coming from suppliers arrive at the cross-dock *simultaneously* from their pickup routes. Such temporal constraints tend to avoid vehicle waiting times at the cross-dock. The problem goal is to minimize the total transportation cost. Time windows were not specified and customer requests must be satisfied within the planning horizon. Moreover, the process of exchanging orders between inbound and outbound vehicles and the temporary storage of some orders at the cross-dock are ignored. Since the problem is NP-hard, the computational efficiency of the MILP approach rapidly deteriorates and a heuristic algorithm based on tabu search had to be developed. Through solving a linear relaxation of the mathematical model, the authors determined a lower bound on the optimal problem value. Good solutions to examples involving up to 50 requests were obtained using the tabu search algorithm within a reasonable amount of time. They featured an average percentage error below 5%.

Wen, Larsen, Clausen, Cordeau, and Laporte (2009) developed a mixed integer programming formulation for the VRPCD problem. In this work, pickup and delivery tasks have predetermined time windows and vehicles coming from suppliers not necessarily arrive at the cross-dock simultaneously. The problem objective is to minimize the total travelled distance. Besides, customer requests are defined in terms of two nodes, namely the pickup node where the freight is loaded and the delivery node to which is destined. Since pickup and delivery operations are carried out at the cross-dock (CD), the CD is represented by four nodes with the first two standing for the starting and ending points of pickup routes, and the last two for the extreme points of delivery routes. In contrast to the formulation of Lee et al. (2006), the exchange operations between pickup and delivery vehicles and, consequently, the temporary storage of some orders at the cross-dock facility are considered. By ignoring the set of constraints linking pickup and delivery tasks in the proposed formulation, it results a relaxed model corresponding to a problem with two independent VRPTW, i.e. a 2-VRPTW problem. The optimal solution to the 2-VRPTW problem provides a lower bound for the VRPCD. To solve large-size problems, a tabu search heuristic embedded within an adaptive memory procedure was proposed. Examples involving up to 200 pairs of nodes were tackled. Non-optimal solutions with objective values less than 5% away from the 2-VRPTW lower bound were found in a short computational time. Recently, Liao, Lin and Shih (2010) presented a new tabu search algorithm for the VRPCD and solved again the set of benchmark problems introduced by Lee et al. (2006). Better solutions were obtained at much less computational time. However, this contribution still assumed that pickup vehicles should arrive at the cross-dock simultaneously.

On the other hand, Dondo, Méndez, and Cerdá (2009) introduced the so-called vehicle routing problem in supply chain management (VRP-SCM). The VRP-SCM problem is a generalization of the N-echelon vehicle routing problem consisting of a number of production facilities, warehouses and destinations. It deals with the transportation of multiple types of products from manufacturer storages to customers either directly or via intermediate warehouses. More important, the allocation of customers to suppliers and the quantities of products shipped from each source to a particular client are problem decision variables. Two or more vehicles can visit a given site to perform pickup and/or delivery operations, and vehicle routes may include several stops at the same site. The solution approach is based on an MILP formulation that relies on a continuous-time representation and applies the global precedence concept for sequencing vehicle stops on every route. However, the approach cannot handle cross-docking operations and incoming shipments from production facilities are stored at the warehouses. Therefore, the model assumes that initial stocks of products available at warehouses are enough to fulfill customer demands during the current planning horizon. Dondo, Méndez, and Cerdá (2011) generalized the MILP approach to allow transshipment operations when initial stocks of some products at warehouses cannot meet the assigned customer demands. In this way, intermediate depots may keep finite stocks of fast-moving products (warehousing) and/or act as cross-dock platforms for slow-moving, high-value items.

3. Problem definition

The vehicle routing with cross-docking is defined as the problem of transporting goods from suppliers to customer locations through a cross-dock facility in order to satisfy a set of customer requests R at minimum transportation cost (see Fig. 1). Each request $r \in R$ includes the shipment size q_r and the Cartesian coordinates of the related pickup and destination nodes. In this way, the distance between pickup/delivery locations of a pair of requests (r, $r1$) $\in R$, given by ($d^P_{r,r1}/d^D_{r,r1}$), and the angular Polar coordinates of the pickup/delivery nodes of request r with the origin at the cross-dock, denoted by (θ^P_r/θ^D_r), can be easily computed. Because the Cartesian coordinates of the cross-dock facility are also problem data, the distance between the cross-dock and the pickup/delivery node of a request r, given by ($d^P_{w,r}/d^D_{w,r}$) can also be determined. Pickup and delivery tasks are accomplished by the same set of homogeneous vehicles with a common capacity Q_v. Subscript v just indicates that Q_v is a vehicle-related parameter.

Each vehicle departs from the cross-dock, serves the assigned pickup/delivery nodes and returns to the terminal for unloading the collected goods on the cross-dock or starting a new pickup

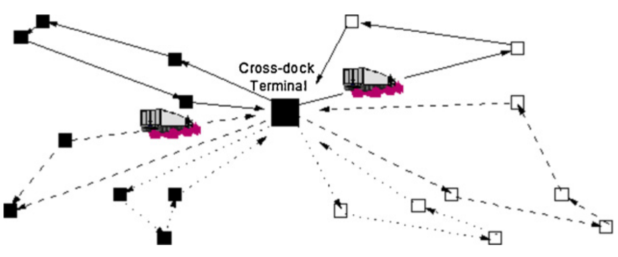

Fig. 1. Illustrating the vehicle routing problem with a single cross-dock terminal.

route. After completing offload operations at the cross-dock, a vehicle can immediately start reloading orders at the shipping door of the intermediate facility for delivering them to their destinations. The service time at each pickup/delivery location has two components: a fixed time part for preparation (ft^P_r/ft^D_r) and a variable component that depends on the size of the load to be picked up or delivered. The loading/unloading rate at each pickup/delivery node is given by (lr_r/ur_r). Similar parameters for the cross-dock are denoted by (ft^P_w/ft^D_w) and (lr_w/ur_w), respectively.

To define pickup and delivery routes, the proposed model includes the following set of binary variables: (a) the assignment variables (YP_{rv}/YD_{rv}), each one allocating a vehicle $v \in V$ to the pickup/delivery node of a request $r \in R$, and (b) the sequencing variables ($XP_{r,r1}/XD_{r,r1}$) establishing the visiting order of pickup/delivery nodes located on the same route. If the pickup nodes of requests (r,$r1$) are visited by the same vehicle and $XP_{r,r1} = 1$, then the request r is served earlier. Otherwise, $XP_{r,r1} = 0$ and the request $r1$ is visited before. If the pickup nodes of requests (r,$r1$) are not in the same route, then the value of $XP_{r,r1}$ is meaningless. In addition, eight sets of continuous variables are incorporated into the proposed formulation to choose: (c) the vehicle arrival time at the pickup/delivery location of every request $r \in R$, given by (T^P_r/T^D_r), (d) the partial travelling cost from the cross-dock to the pickup/delivery node of each request r, (C^P_r/C^D_r), and (e) the overall travelling time (AT^P_v/AT^D_v) and travelling cost (OC^P_v/OC^D_v) for the pickup/delivery route assigned to vehicle v. Indeed, AT^D_v represents the total service time of vehicle v. The values of (OC^P_v/OC^D_v) are directly related to the total distance travelled by vehicle v along the assigned pickup/delivery route. In the model, the parameter uc_v stands for the travel cost per unit distance and the coefficient sp_v is the average speed of vehicle v that is equal for every truck.

The other variables included in the model are related to the cross-dock truck scheduling. The most important time events at the cross-dock are the times at which inbound vehicles finish unloading operations and become ready to start doing delivery tasks. The ready time for vehicle v, given by RT^P_v, represents the earliest time at which it can start loading the orders for delivery to their destinations. Besides, it stands for the earliest time at which the orders taken out of vehicle v at the receiving dock can be loaded onto outbound trucks for delivery. Therefore, there will occur as many events $n \in N$ at the cross-dock as the number of inbound trucks, where N is a preordered set of events whose element n takes place before event ($n+1$). Moreover, each event n is associated to one inbound truck and vice versa. To pair events and pickup vehicles, a set of 0–1 assignment variables WP_{nv} is defined. When $WP_{nv} = 1$, the pickup vehicle v is assigned to event n and the event time TE_n is then defined by the ready time of truck v ($TE_n = RT^P_v$). In this way, the model can trace the accumulated set of orders that has been unloaded at the cross-dock up to event n given by the variable IT_{rn}. On the other hand, a particular outbound vehicle v can be assigned to event n only if, at time TE_n, all the orders to be loaded on truck v for delivery are already available at the cross-dock (i.e. the value of IT_{rn} should be q_r if the loading of order r onto some vehicle is started at time TE_n). To this end, the proposed model also includes the assignment variables WD_{nv} allocating outbound vehicles to events. In this case, several delivery trucks can be assigned to the same event.

On the other hand, the model constraints can be grouped into four categories: (1) pickup route building constraints assigning vehicles to pickup nodes and sequencing requests on the same tour; (2) delivery route building constraints allocating vehicles to delivery nodes and sequencing requests on the same delivery route; (3) vehicle capacity constraints; and (4) cross-dock truck scheduling constraints pairing inbound/outbound vehicles to cross-dock events and tracing the accumulated stock of cargoes available on the cross-dock at each time event.

4. Model assumptions

The proposed mathematical formulation for the integrated VRPCD problem is based on the following assumptions:

(i) A set of identical vehicles are used to transport goods from suppliers to retailers through a cross-dock facility.
(ii) A single cross-dock with a sufficiently large number of receiving/shipping dock doors is available so that inbound/outbound vehicles can immediately start unloading or loading operations after arrival without delay.
(iii) All the vehicles are available at the cross-dock from the start of the horizon.
(iv) The same fleet of vehicles sequentially carries out the required pickup and delivery tasks.
(v) The whole process must be completed within the planning horizon.
(vi) The amounts of goods to be loaded/unloaded at supply/delivery nodes are known.
(vii) Split deliveries are not allowed and each pickup point (supplier) and each delivery point (retailer) must be visited by a single vehicle only once.
(viii) Vehicles can serve more than one supplier or customer.
(ix) Pickup and delivery routes start and end at the cross-dock.
(x) The total quantity of goods in a vehicle must never exceed its capacity.
(xi) The service time at supply and delivery nodes is the sum of two components: a fixed part (ft_r^P/ft_r^D) and a variable service-time contribution, with the later one directly increasing with the size of the cargo to be loaded or unloaded.
(xii) Goods picked up and delivered by the same vehicle are not unloaded at the cross-dock.
(xiii) The total amount of goods unloaded at the receiving dock and loaded onto delivery vehicles in the shipping dock should be equal, i.e. there is no end inventory at the cross-dock.

5. The Milp mathematical model

5.1. Pickup route building constraints

5.1.1. Allocating pickup requests to vehicles

Every pickup request should be allocated to exactly one vehicle. If $YP_{rv} = 1$, then the pickup site of request r is served by the inbound vehicle v. Idle vehicles have no assigned pickup requests at all.

$$\sum_{v \in V} YP_{rv} = 1 \quad \forall r \in R \tag{1}$$

5.1.2. Pickup routing cost sequencing constraints

The mathematical formulation of the sequencing constraint set is based on the notion of global precedence. It uses a single binary variable $XP_{r,r1}$ to choose the relative visiting order of two request sites $r,r1 \in R$ located on the same inbound route. Just the variable $XP_{r,r1}$ with $r < r1$ is defined for the pair $(r,r1)$. The proposed sequencing constraints allow to determine the partial travel cost up to every visited location on each inbound route.

The partial travelling cost from the cross-dock platform to the first served pickup site. Eq. (2) provides a lower bound on the pickup travel cost for the first leg of the vehicle trip. The parameter uc_v stands for the unit travel cost and $d_{w,r}^P$ represents the distance between the cross-dock facility, identified by the subscript w, and the pickup site of request r.

$$C_r^P \geq uc_v d_{w,r}^P YP_{rv}, \quad \forall r \in R, v \in V \tag{2}$$

The partial travel cost from the start up to the pickup site of request r. Eqs. (3a) and (3b) relate the partial travel costs up to the pickup sites of a pair of requests $r,r1 \in R$ only if both are visited by the same vehicle v ($YP_{rv} = YP_{r1,v} = 1$). Let C_r^P be the partial travel cost from the start up to the pickup location of request r. If $XP_{r,r1} = 1$ ($r < r1$), then C_{r1}^P must be larger than C_r^P by at least the travel cost along the path directly connecting both locations, i.e. the shortest route between r and $r1$. Otherwise, $XP_{r,r1} = 0$ and C_{r1}^P should be lower than C_r^P by at least $uc_v d_{r1,r}^P$. The parameter M_C^P stands for an upper bound on the value of C_r^P for any request r.

$$C_{r1}^P \geq C_r^P + uc_v d_{r,r1}^P - M_C^P(1 - XP_{r,r1}) - M_C^P(2 - YP_{r,v} - YP_{r1,v}) \quad \forall r, r1 \in R(r < r1), v \in V \tag{3a}$$

$$C_r^P \geq C_{r1}^P + uc_v d_{r,r1}^P - M_C^P XP_{r,r1} - M_C^P(2 - YP_{r,v} - YP_{r1,v}) \quad \forall r, r1 \in R(r < r1), v \in V \tag{3b}$$

The overall travel cost along the entire pickup route. The vehicle pickup trip should end at the cross-dock facility. The right-hand side of Eq. (4) provides a lower bound on the total travel cost incurred in completing the vth-vehicle tour. The largest bound determining the value of OC_v^P is set by the last pickup location visited by vehicle v.

$$OC_v^P \geq C_r^P + uc_v d_{r,w}^P - M_C^P(1 - YP_{r,v}), \quad \forall r \in R, v \in V \tag{4}$$

5.1.3. Pickup travel time sequencing constraints

These constraints allow determining the elapsed travel time to go from the cross-dock to any pickup location on the same inbound route.

5.1.3.1. The travel time from the cross-dock to the first-served pickup location. Eq. (5) sets a bound on the pickup travel time for the first leg of the vehicle trip. The parameter sp_v stands for the average vehicle speed and d_{wr}^P is the distance between the cross-dock facility w and the pickup site of request r.

$$T_r^P \geq \left(\frac{d_{wr}^P}{sp_v}\right) YP_{rv}, \quad \forall r \in R, v \in V \tag{5}$$

5.1.3.2. The partial travel time from the cross-dock up to the pickup site of request r. Constraints (6a) and (6b) link the travel times spent to reach the pickup sites of a pair of requests $r,r1 \in R$ in case both locations are visited by the same vehicle v ($YP_{rv} = YP_{r1,v} = 1$). The service time at every site is assumed to be the sum of two components: a fixed (ft_r^P) and a variable service-time contribution, with the later one directly increasing with the size of the freight to load on the vehicle. The parameter lr_r stands for the shipment loading rate at the pickup site of request r. In turn, M_T^P is an upper bound on the value of T_r^P for any request r.

$$T_{r1}^P \geq T_r^P + ft_r^P + lr_r q_r + \left(\frac{d_{r,r1}^P}{sp_v}\right) - M_T^P(1 - XP_{r,r1}) - M_T^P(2 - YP_{r,v} - YP_{r1,v}) \quad \forall r, r1 \in R(r < r1),\ v \in V \tag{6a}$$

$$T_r^P \geq T_{r1}^P + ft_{r1}^P + lr_{r1} q_{r1} + \left(\frac{d_{r1,r}^P}{sp_v}\right) - M_T^P XP_{r,r1} - M_T^P(2 - YP_{r,v} - YP_{r1,v}) \quad \forall r, r1 \in R(r < r1),\ v \in V \tag{6b}$$

5.1.3.3. Overall travel time along the entire pickup route. The arrival time of vehicle v at the cross-dock (AT_v^P) can be obtained by adding the following items to the visiting time of vehicle v at the last served pickup site: (i) the service time at the last visited location; and (ii) the travel time along the return leg to the cross-dock platform. Since the last pickup node is not known beforehand, Eq. (7) is written for all requests allocated to vehicle v. The request providing the largest RHS defines the value of AT_v^P.

$$AT_v^P \geq T_r^P + ft_r^P + lr_r q_r + \left(\frac{d_{r,w}^P}{sp_v}\right) - M_{TP}(1 - YP_{r,v}) \quad \forall r \in R, v \in V \quad (7)$$

5.1.4. Requests with pickup and delivery tasks both served by the same vehicle

If pickup and the delivery sites for request r are both served by the same vehicle, the related transshipment operations at the cross-dock are no longer required. In other words, the freight previously loaded at the pickup location is never removed from the vehicle at the cross-dock platform. If the binary variable YD_{rv} indicates the allocation of the delivery site of request r also to vehicle v, then a request r fully served by vehicle v is characterized by: $YR_{rv} = YD_{rv} = 1$. Recognizing fully served requests by a particular vehicle is very important for an exact calculation of the vehicle service times. The duration of the related unloading operations for requests fully served by the same vehicle should obviously be ignored. Let YR_{rv} be a non-negative continuous variable with a domain [0,1] that is used to identify fully-served requests by vehicle v whenever is equal to one. Eqs. (8)–(10) drives YR_{rv} to one whenever $YP_{rv} = YD_{rv} = 1$, and drops YR_{rv} to zero if either of such variables (YP_{rv} or YD_{rv}) are null.

$$YR_{r,v} \leq YP_{r,v} \quad \forall v \in V, r \in R \quad (8)$$

$$YR_{r,v} \leq YD_{r,v} \quad \forall v \in V, r \in R \quad (9)$$

$$YR_{r,v} \geq YP_{r,v} + YD_{r,v} - 1 \quad \forall v \in V, r \in R \quad (10)$$

5.1.5. Vehicle release times to start serving delivery requests

Let RT_v^P be the earliest time at which vehicle v completes its pickup duties and can start performing delivery activities, i.e. unloading operations at the cross-dock have finished. A lower bound on the value of RT_v^P is provided by Eq. (11). It is obtained by adding the length of the unloading operations at the cross-dock center to the arrival time AT_v^P. In Eq. (11), the parameter ur_w represents the shipment unloading rate at the cross-dock.

$$RT_v^P \geq AT_r^P + ft_w^P + ur_w \left[\sum_{r \in R} q_r(YP_{r,v} - YR_{r,v})\right] \quad \forall r \in R, \; v \in V \quad (11)$$

5.1.6. Optional speed-up constraints relating vehicle release times and routing costs

To speed up the convergence rate to the best solution, Eq. (12) directly relating the release time RT_v^P and the total routing cost for vehicle v may optionally be considered.

$$RT_v^P \geq \left(\frac{OC_v^P}{uc_v \, sp_v}\right) + \sum_{r \in R}(ft_r^P + lr_r q_r) YP_{rv} + ft_w^P + ur_w \left[\sum_{r \in R} q_r(YP_{r,v} - YR_{r,v})\right] \quad \forall v \in V \quad (12)$$

5.2. Vehicle capacity constraints

The total load transported by a pickup vehicle cannot exceed its maximum capacity given by Q_v.

$$\sum_{r \in R} q_r \, YP_{rv} \leq Q_v \quad \forall v \in V \quad (13)$$

5.3. Time events at the cross-dock facility

5.3.1. Allocating events to pickup vehicle release times

An event n occurs at the cross-dock whenever a pickup vehicle v arrives at this facility, and completely unloads the shipment to be delivered by other vehicles. The related event time TE_n represents the release time of vehicle v from pickup assignments. In other words, TE_n is the earliest time at which vehicle v can start loading the assigned shipments (previously picked up by other vehicles) for delivering them to their destinations. If TE_n is the release time of vehicle v, then the 0–1 variable WP_{nv} allocating time events to vehicles will be equal to 1. There should be defined as many events at the cross-dock as the number of available vehicles, i.e., $|N| = |V|$. Events assigned to unused vehicles will never occur. They are fictitious events that may arise only if vehicle fixed costs are considered and the number of used vehicles becomes a problem variable. Eq. (14) states that an inbound vehicle must be exactly assigned to a single event. Reciprocally, according to Eq. (15), an event should be exactly assigned to a single inbound vehicle. If $WP_{n,v}$ is equal to one, then $TE_n = RT_v^P$. In the general case, the value of RT_v^P varies with the vehicle v. Therefore, the events at the cross facility will generally occur at different times.

$$\sum_{n \in N} WP_{n,v} = 1 \quad \forall v \in V \quad (14)$$

$$\sum_{v \in V} WP_{n,v} = 1 \quad \forall n \in N \quad (15)$$

5.3.2. Preordering events occurring at the cross-docking platform

Through Eq. (16), the events are assigned to inbound vehicles in the same order that they complete their pickup duties.

$$TE_{n1} \geq TE_n \quad \forall n, n1 \in N (n < n1) \quad (16)$$

5.3.3. Relating vehicle release times and event times

If event n has been assigned to vehicle v ($WP_{nv} = 1$), then $TE_n = RT_v^P$. Through Eq. (17), the value of RT_v^P is imposed as a lower bound for TE_n whenever $WP_{n1,v} = 1$ (for any $n1 \leq n$). The equality condition is forced by Eqs. (18)–(20).

$$TE_n \geq RT_v^P - M_T^P(1 - WP_{n1,v}) \quad \forall n, n1 \in N(n1 \leq n), v \in V \quad (17)$$

$$TE_n \leq RT_v^P \quad \forall v \in V, n = first(N) \quad (18)$$

$$RT_v \leq TE_{n1} + M_T^P(1 - WP_{n,v}) \quad \forall n, n1 \in N(n1 \geq n), v \in V \quad (19)$$

$$\sum_{n \in N} TE_n = \sum_{v \in V} RT_v^P \quad (20)$$

5.4. Accumulated unloaded orders at the cross-dock up to each time event

Let the variable I_{rnv} denote that the cargo associated to request r and transported by vehicle v is available in the cross-dock at time $t = T_n$ if I_{rnv} is equal to q_r. Variable I_{rnv} can take one of two possible values: 0 (not available) or q_r (available). As stated by Eqs. (21) and (22), the value of I_{rnv} will be driven to zero if either vehicle v does not pickup the rth-cargo ($YP_{r,v} = 0$) or vehicle v serves the pickup location of request r but does not still complete the unloading operations at time T_n ($WP_{n,v} = 0$). If instead ($YP_{r,v} + WP_{n,v}$) = 2, Eqs. (22) and (23) make I_{rnv} equal to q_r.

$$I_{rnv} \leq q_r WP_{n,v} \quad \forall r \in R, n \in N, v \in V \quad (21)$$

$$\sum_{n \in N} I_{rnv} \leq q_r YP_{r,v} \quad \forall r \in R, v \in V \quad (22)$$

$$I_{rnv} \geq q_r(WP_{n,v} + YP_{rv} - 1) \quad \forall n \in N, r \in R, v \in V \quad (23)$$

On the other hand, IT_{rn} will be equal to q_r if the rth-cargo is available in the cross-dock facility at time T_n no matter what vehicle makes the pickup operation. The rth-cargo will be available at the cross-dock if unloading operations from an incoming trailer at the cross-dock finish at time T_n or at previous time events T_{n1} ($n1 < n$). This condition is ensured by Eq. (24).

$$IT_{rn} = \sum_{\substack{n1 \in N \\ n1 \le n}} \sum_{v \in V} I_{r,n1,v} \quad \forall r \in R, n \in N \tag{24}$$

5.5. Allocating time events at the cross-dock to outbound vehicles

Let the binary variable $WD_{n,v}$ denotes the assignment of time event n to delivery vehicle v whenever $WD_{n,v} = 1$. If so, truck v can start loading the assigned freights for delivery at time TE_n. Eq. (25) states that a single event should be assigned to every delivery vehicle, but more than one outbound truck can be allotted to a particular event. Obviously, the outbound vehicle v cannot be allocated to event n if its release time RT_v from pickup duties is greater than TE_n. Then, Eq. (26) drives $WD_{n,v}$ to zero if $WD_{n1,v}$ is equal to 1 for some $n1 > n$.

$$\sum_{n \in N} WD_{n,v} = 1 \quad \forall v \in V \tag{25}$$

$$WD_{n,v} \le \sum_{\substack{n1 \in N \\ n1 \le n}} WP_{n1,v} \quad \forall n \in N, v \in V \tag{26}$$

5.6. Outbound route building constraints

5.6.1. Allocating delivery requests to vehicles

Each delivery request should be exactly allocated to a single vehicle. In Eq. (27), the binary variable YD_{rv} stands for the allocation of the delivery request r to trailer v whenever $YD_{rv} = 1$.

$$\sum_{v \in V} YD_{rv} = 1 \quad \forall r \in R \tag{27}$$

5.6.2. Constraints on delivery allocations

The rth-cargo can be assigned to vehicle v and the related delivery operations can start at time TE_n only if vehicle v is already released from pickup duties and the rth-cargo is available in the cross-dock facility both at time TE_n. If such conditions hold, $WD_{n,v} + YD_{rv} = 2$ and $IT_{r,n}$ is equal to q_r, Eq. (28) allows to choose $WD_{n,v} = YD_{rv} = 1$ only if $IT_{r,n} \ge q_r$.

$$IT_{rn} \ge q_r(WD_{n,v} + YD_{rv} - 1) \quad \forall n \in N, r \in R, v \in V \tag{28}$$

5.6.3. Outbound routing cost sequencing constraints

The mathematical formulation of outbound routing cost sequencing constraints are quite similar to those proposed in Section 5.1.2 for inbound routing cost sequencing restraints. The mathematical structure remains the same but the name and meaning of variables and parameters involved in these new constraints are obviously different. Sequencing variables for outbound routes establishing the visiting order of delivery locations are represented by $XD_{r,r1}$, the partial delivery routing cost up to the delivery location of request r is given by C_r^D, and the total cost for the delivery trip of vehicle v is denoted by OC_v^D. Moreover, $d_{r,r1}^D$ denotes the travel distance between the delivery locations of requests r and $r1$, and $d_{w,r}^D$ represents the distance between the unique cross-dock w and the delivery location of request r. Besides, ft_r and ur_r stand for the fixed part of the service time and the unloading rate of the cargo at the delivery location of request r, respectively. Sequencing constraints providing the outbound vehicle routing costs are given by Eqs. (29)–(32).

Partial travel cost from the cross-dock to the first served delivery site.

$$C_r^D \ge uc_v d_{w,r}^D YD_{rv}, \quad \forall r \in R, v \in V \tag{29}$$

Partial travel cost from the cross-dock up to the delivery site of request r.

$$C_{r1}^D \ge C_r^D + uc_v \; d_{r,r1}^D - M_C^D \; (1 - XD_{r,r1}) - M_C^D \; (2 - YD_{r,v} - YD_{r1,v}) \quad \forall r, r1 \in R (r < r1), v \in V \tag{30}$$

$$C_r^D \ge C_{r1}^D + uc_v \; d_{r,r1}^D - M_C^D \; XD_{r,r1} - M_C^D \; (2 - YD_{r,v} - YD_{r1,v}) \quad \forall r, r1 \in R (r < r1), v \in V \tag{31}$$

Overall vehicle travelling cost along the entire delivery route.

$$OC_v^D \ge C_r^D + uc_v \; d_{r,w}^D - M_C^D(1 - YD_{r,v}), \quad \forall r \in R, v \in V \tag{32}$$

5.6.4. Delivery travel time constraints

The mathematical structure of delivery travel time sequencing constraints are similar to the ones proposed in Section 5.1.3 for pickup travel time sequencing restraints. Both sets share similar sequencing variables but the name, now called $XD_{r,r1}$, and the meaning of variables and parameters obviously change. The partial delivery travel time up to the delivery location of request r is given by T_r^D, and the end time of the delivery trip for vehicle v is represented by AT_v^D. Besides, ft_w^D represents the average time required to move goods from the inbound to the outbound dock, while lr_w stand for the loading rate of goods onto delivery vehicles. Sequencing constraints providing the outbound vehicle travel times are given by Eqs. (33)–(36). The optional speed-up constraints (37) and (38) set bounds on the value of the arrival time AT_v^D based on travelling costs and the times related to the first and last events taking place at the cross-dock.

Vehicle travel time from the cross-dock facility to the first-served delivery location

$$T_r^D \ge TE_n + ft_w^D + lr_w \left[\sum_{r \in R} q_r (YD_{r,v} - YR_{r,v}) \right] + \left(\frac{d_{r,w}^D}{sp_v} \right) - M_T^D (2 - WD_{n,v} - YD_{r,v}) \quad \forall n \in N, r \in R, v \in V \tag{33}$$

Partial travel time from the start up to the pickup site of request r

$$T_{r1}^D \ge T_r^D + ft_r^D + ur_r q_r + \left(\frac{d_{r,r1}^D}{sp_v} \right) - M_T^D (1 - XD_{r,r1}) - M_T^D (2 - YD_{r,v} - YD_{r1,v}) \quad \forall r, r1 \in R (r < r1), v \in V \tag{34}$$

$$T_r^D \ge T_{r1}^D + ft_{r1}^D + ur_{r1} q_{r1} + \left(\frac{d_{r1,r}^D}{sp_v} \right) - M_T^D XD_{r,r1} - M_T^D (2 - YD_{r,v} - YD_{r1,v}) \quad \forall r, r1 \in R (r < r1), v \in V \tag{35}$$

Vehicle arrival times at the cross-dock after serving the assigned delivery nodes

$$AT_v^D \ge T_r^D + ft_r^D + ur_r q_r + \left(\frac{d_{r,w}^D}{sp_v} \right) - M_T^D (1 - YD_{r,v}) \quad \forall r \in R, v \in V \tag{36}$$

5.6.5. Optional speed-up constraints on vehicle arrival times

They directly relate delivery travel times, delivery travel costs and time events at the cross-dock terminal.

$$AT_v^D \geq TE_{first(N)} + \left(\frac{OC_v^D}{uc_v\, sp_v}\right) + ft_w^D + lr_w \left[\sum_{r \in R} q_r(1 - YR_{rv})\right] + \sum_{r \in R}(ft_r^D + ur_r q_r)YD_{rv} \quad \forall v \in V \tag{37}$$

$$AT_v^D \leq TE_{last(N)} + \left(\frac{OC_v^D}{uc_v\, sp_v}\right) + ft_w^D + lr_w \left[\sum_{r \in R} q_r(1 - YR_{rv})\right] + \sum_{r \in R}(ft_r^D + ur_r q_r)YD_{rv} \quad \forall v \in V \tag{38}$$

5.7. Outbound vehicle capacity constraints

$$\sum_{r \in R} q_r YD_{rv} \leq Q_v \quad \forall v \in V \tag{39}$$

5.8. Alternative objective functions

Two alternative objective functions have been adopted. The first one given by Eq. (40) aims to minimize the total vehicle routing cost while the other modelled by Eq. (41) looks for the minimum makespan.

$$\min \quad z = \sum_{v \in V}(OC_v^P + OC_v^D) \tag{40}$$

$$\min \quad z = H \quad \text{with} \quad H \geq AT_v^D, \quad v \in V \tag{41}$$

Therefore, the proposed exact formulation (EF) for the VRPCD problem comprises the set of constraints (1)–(39), and Eq. (40) or (41) as the objective function.

6. Sweep-based approach for vehicle allocation

The mathematical model presented in Section 5 shows a sharp increase of the solution time with the number of customer requests. A significant fraction of the computational cost is associated to the process of assigning vehicles to pickup/delivery nodes. In order to get a more efficient VRPCD approach, the sweep-heuristic algorithm proposed by Gillett and Miller (1974) to allocate requests to vehicles has been modelled and used in combination with the VRPCD formulation of Section 5. The sweep-rule algorithm is a heuristic technique that efficiently solves VRP problems. It assumes a unique depot and a fleet of vehicles all having the same capacity. The depot is at the origin of a polar coordinate system through which each pickup/delivery location is described in terms of two coordinates: the radial (r_r) and the angular (θ_r) coordinates. For the vehicle assignment process, the customer nodes are arranged by increasing angular coordinates. Following this order, nodes are assigned to the current vehicle while it is not overloaded. Otherwise, a new vehicle is chosen and the procedure is continued until every node has been assigned to exactly one vehicle.

There is also a geometrical way of describing the sweep-heuristic algorithm. A ray drawn from the depot initially featuring a zero angular coordinate is swept in a clockwise direction. The rotation continues while the aggregate demand at the nodes swept by the ray (i.e. the assigned locations) does not exceed the capacity of the current vehicle. Otherwise, the ray rotation and the assignment of customers to the current vehicle are both stopped. A new vehicle is considered and the ray rotation starts for a second time. The procedure ends when the ray reaches again the initial angular coordinate and the whole region to be served has been swept. In this way, customer locations are grouped into a number of circular sectors each one assigned to a different vehicle. Though the quality of the node-vehicle assignment process is deteriorated, the procedure can still be applied even if a heterogeneous fleet is considered. In that case, vehicles will be ordered according their priorities and the one currently at the top of the list is next considered for customer allocation. Vehicles with larger capacities may feature higher priorities. The proposed mathematical model even improves the classical sweep-heuristic algorithm by choosing the best starting polar angle. In other words, the initial ray position is no longer a problem datum but a problem variable whose value is optimized. Since the mathematical formulation for both pickup and delivery routing problems are equivalent, the proposed set of assignment constraints is only presented for pickup routes but an identical one should be written for delivery tours. In the proposed formulation, the continuous variable ϕ_v^P will stand for the lower angular limit of v th-pickup sector, while the variable $\Delta\phi_v^P$ denotes the angular width of such a zone.

6.1. Angular limits and width of the vth-circular sector ($\Delta\phi_v^P$)

As stated by Eq. (42), the upper angular limit of sector v is the lower limit of sector $(v+1)$. Moreover, the set of zones defined by the model should cover the entire region to be served. By Eq. (43), the sum of their angular widths must be equal to 2π.

$$\phi_{v+1}^P = \phi_v^P + \Delta\phi_v^P \quad \forall v \in V(v < |V|) \tag{42}$$

$$\sum_{v \in V} \Delta\phi_v^P = 2\pi \tag{43}$$

6.2. Unused sectors arising first in the set V

A number of zones at least equal to the number of available vehicles should be predefined. However, some of these zones could be fictitious because not all the vehicles are used. This could happen only if fixed vehicle costs are considered and the number of used trucks is a problem variable. The binary variable U_v^P denoting the existence of sector v has a zero value for a fictitious zone. Constraint (44) drives the angular width of any fictitious sector to zero. On the other hand, Eq. (45) ensures that fictitious sectors, if any, will arise first. Variables U_v^P allow considering vehicle fixed costs in the objective function.

$$\Delta\phi_v^P \leq 2\pi U_v^P \quad \forall v \in V \tag{44}$$

$$U_{v+1}^P \geq U_v^P \quad \forall v \in V(v < |V|) \tag{45}$$

6.3. Allocating nodes to vehicles

Each pickup location should be assigned to exactly one vehicle. This condition has already been considered through Eq. (1). If the sweep algorithm is not applied, Eq. (1) will be the only one arising at the VRPCD model to allocate customer locations to vehicles. Otherwise, the set of constraints presented in this section should also be considered. If vehicle v is not used ($U_v^P = 0$), then Eq. (46) does not allow to assign it customer locations.

$$\begin{aligned} &\sum_{v \in V} YP_{rv} = 1 \quad \forall r \in R \\ &YP_{rv} \leq U_v^P \quad \forall r \in R, v \in V \end{aligned} \tag{46}$$

6.4. Feasible allocation of nodes to the circular sector v

For every zone before the last one, all pickup locations featuring an angular coordinate θ_r^P within the sector v, $\theta_r^P \in \left[\phi_v^P, \phi_{v+1}^P\right]$, can be allocated to vehicle v. This condition is enforced by Eqs. (47) and (48). The vehicle assignment for locations just on the boundary between sectors v and $v+1$ is left to the model.

$$\phi_v^P \le \theta_r^P + 2\pi(1 - YP_{rv}) \quad \forall r \in R, v \in V(v < \left|V\right|) \tag{47}$$

$$\phi_{v+1}^P \ge \theta_r^P YP_{rv} \quad \forall r \in R, v \in V(v < \left|V\right|) \tag{48}$$

6.5. Allowing the first used sector to start at the best angular location

The last zone receives a special treatment because the rotating ray may start its movement from an initial polar angle ϕ_1^P greater than $(\min_{r \in R} \theta_r^P)$ Recall that the start polar angle ϕ_1^P is chosen by the model. Then, pickup locations with an angular coordinate $\theta_r^P \in \left[0, \phi_1^P\right)$ must be allocated to the last sector $v = |V|$. Let us define the continuous variable ξ_r^P that takes a value equal to one whenever the pickup location of request r satisfies the condition: $\theta_r^P \in \left[0, \phi_1^P\right)$. If so, Eq. (49) assigns request r to the last sector.

$$YP_{rv} \ge \xi_r^P \quad \forall r \in R, v = \left|V\right| \tag{49}$$

Moreover, Eq. (50) reduces to $\theta_r^P \xi_r^P \le \phi_v^P$, $v \in V$ for every existent sector v. If $\xi_r^P = 1$, then $\theta_r^P \le \phi_v^P$ for any $v \in V$. Otherwise, constraint (50) becomes redundant.

$$\theta_r^P(\xi_r^P + U_v^P - 1) \le \phi_v^P \quad \forall r \in R, v \in V \tag{50}$$

Fictitious zones arise first and their angular widths are equal to zero. Then, the lower angular limit of the first existent sector v_n will always be equal to $\phi_n^P = \phi_{n-1}^P = \cdots = \phi_1^P$. Then, the constraint (51) can be incorporated in the problem formulation to speed up the rate of convergence to the optimal solution.

$$\theta_r^P \ge \phi_v^P - 2\pi\xi_r^P \quad \forall r \in R, v = 1 \tag{51}$$

Besides, Eq. (52) replaces Eq. (47) for the last sector. This constraint forcing a request assigned to sector $v = |V|$ to have an angular coordinate $\theta_r^P \ge \phi_{|V|}^P$ no longer applies if ξ_r^P is equal one. When $\xi_r^P = 0$, Eq. (52) looks similar to Eq. (47).

$$\phi_v^P \le \theta_r^P + 2\pi(1 + \xi_r^P - YP_{rv}) \quad \forall r \in R, v = \left|V\right| \tag{52}$$

If the sweep algorithm is used to allocate vehicles to pickup/delivery nodes, the problem model will include the set of Eqs. (1)–(39), the sweep-heuristic constraints (42)–(52) for pickup and delivery routes, and the objective function (40) or (41).

6.6. The modified sweep-based approach for vehicle allocation

The sweep-based approach only considers solutions with non-overlapping routes, i.e. each vehicle route belongs to a different angular sector. When time windows are specified for the start of pickup or delivery tasks, vehicle routes may no longer have the tear-drop shape and some overlapping between nearby tours can also arise. Sometimes, all solutions with non-overlapping routes become infeasible if the number of vehicles is not increased. To get feasible or better solutions with the SBF, there are two types of remedies: (a) increasing the number of available vehicles and angular sectors; (b) keeping the same number of vehicles and, at the same time, increasing the size of the feasible region by also accounting for solutions with overlapping routes. To face VRPCD problems with time windows, a modified sweep-based approach that admits tour overlapping has also been developed. It consists of defining a maximum angular overlapping $\Delta\theta$ between adjacent sectors to allow the assignment of nodes to zone v even though they belong to adjacent zones $(v+1)$ or $(v-1)$. In addition, new continuous variables (DV_r^P/DV_r^D) are added to the model whose values cannot exceed the maximum overlapping $\Delta\theta$ as specified by Eq. (53). The value of (DV_r^P/DV_r^D) will take a value greater than zero just for those pickup/delivery nodes allocated to adjacent sectors. The new expressions for Eqs. (47), (48), (51) and (52) are given by Eqs. (54)–(57). Similar variables and constraints for the delivery routes should be included in the formulation.

$$DV_r^P \le \Delta\theta^P \quad \forall r \in R \tag{53}$$

$$\phi_v^P \le \theta_r^P + DV_r^P + 2\pi(1 - YP_{rv}) \quad \forall r \in R, v \in V(v < \left|V\right|) \tag{54}$$

$$\phi_{v+1}^P \ge \theta_r^P YP_{rv} - DV_r^P \quad \forall r \in R, v \in V(v < \left|V\right|) \tag{55}$$

$$\theta_r^P + DV_r^P \ge \phi_v^P - 2\pi\xi_r^P \quad \forall r \in R, v = 1 \tag{56}$$

$$\phi_v^P \le \theta_r^P + DV_r^P + 2\pi(1 + \xi_r^P - YP_{rv}) \quad \forall r \in R, v = \left|V\right| \tag{57}$$

7. Results and discussion

To illustrate the potential of the proposed sweep-based VRPCD formulation on providing high-quality solutions with a remarkable computational efficiency, an extensive number of VRPCD examples have been studied. The sweep-based approach was first validated by solving a series of small-to-medium size problems and comparing the results obtained with those found using the exact formulation (EF). After that, it was applied to larger problem instances involving up to 50 customer requests. The least total transportation cost and the minimum makespan were alternatively selected as the problem objective for those examples.

The optimal solution or the best outcome within the CPU time limit together with the solution obtained after 1000 CPU seconds are reported. All problem instances were run on a 2.66 MHz six core dual processor PC with 24 MB RAM. Table 1 presents the data for 50 transportation requests, including the shipment size, the Cartesian coordinates of the related pickup and delivery nodes and the service time windows for each customer request. All the problem instances were generated by considering the first N requests of Table 1, with the parameter N varying from 10 to 50. Each example is referred to by the quantity of requests (N) and the number of available vehicles. The vehicle capacity for each example is given in Table 2 and the selected parameter values for all problem instances were: $ft_r^P = ft_r^D = 0.5$; $lr_r = ur_r = 0.2$; $ft_w^P = ft_w^D = 0.5$; $lr_w = ur_w = 0.5$; $uc_v = 1$, and $sp_v = 1$. The customer orders should be satisfied within the planning horizon going from $t = 0$ to $t = tv_v^{\max} = 400$ h. A relative gap tolerance of 10^{-3} or a CPU time limit of 3600 s has been chosen as the stopping criterion. Exceptionally, the time limit was increased to 10,800 s. For $N > 30$, the gap optimality was set at 10^{-2}.

7.1. Validation of the sweep-based formulation (SBF)

Table 3 shows the best solutions for a set of six examples involving 10–15 requests using the exact formulation and the sweep-based model, respectively. The customer orders are served by 2 or 3 vehicles depending on the problem size, and the problem goal is the minimization of the total transportation cost.

With the exception of example 11R-2V, the sweep-based model practically provides near-optimal solutions for every problem instance in a short CPU time. For a pair of examples, it even provides better solutions than the exact formulation within the CPU time limit of 10,800 s. From Table 3, it follows that the solution time is decreased by a factor much higher than 50 when using the sweep-based formulation. For examples with $N \ge 12$, the exact approach has not converged after a CPU time of 10,800 s and the relative gap

Table 1
Data for the examples involving up to 50 transportation requests.

Request	Pick-up locations		Load	Pickup time windows		Delivery locations		Delivery time windows	
	X coord	*Y* coord		*a*	*b*	*X* coord	*Y* coord	*a*	*b*
r1	41	49	10	30	100	20	20	200	320
r2	35	17	7	0	100	31	52	190	330
r3	55	45	13	20	60	24	12	240	310
r4	55	20	19	0	60	35	40	300	350
r5	15	30	26	20	60	41	37	260	290
r6	25	30	3	0	60	53	52	200	280
r7	20	50	5	20	80	45	30	180	230
r8	10	43	9	20	100	40	25	260	300
r9	55	60	16	70	200	11	14	220	240
r10	30	60	16	0	80	65	7	220	270
r11	20	42	12	80	130	60	12	280	320
r12	50	35	19	50	100	13	52	200	300
r13	30	25	23	0	40	63	65	260	300
r14	15	10	20	30	100	47	47	260	300
r15	30	5	8	0	60	40	60	200	340
r16	10	20	19	20	60	65	55	220	280
r17	5	30	2	0	100	64	42	260	350
r18	20	40	12	90	130	23	3	250	300
r19	15	60	17	80	140	5	5	230	270
r20	45	65	9	0	100	8	56	260	300
r21	45	20	11	80	120	6	68	280	340
r22	45	10	18	0	80	35	69	240	320
r23	55	5	29	10	80	2	48	260	300
r24	44	22	12	0	100	25	50	200	400
r25	28	25	8	40	80	46	39	230	275
r26	40	47	15	10	80	22	39	220	260
r27	48	23	22	0	40	31	33	200	340
r28	26	29	7	0	100	50	20	270	320
r29	18	22	11	40	80	18	43	220	280
r30	45	38	8	20	80	50	29	250	320
r31	53	43	14	0	100	18	15	200	350
r32	40	19	9	0	100	27	42	200	400
r33	29	51	17	20	100	60	41	210	280
r34	20	36	12	30	100	39	22	200	400
r35	50	25	14	50	100	45	42	170	300
r36	67	19	10	30	80	37	85	220	275
r37	16	24	17	0	100	71	8	200	330
r38	47	85	6	0	100	17	83	220	270
r39	21	66	21	20	80	5	74	250	290
r40	74	31	14	30	70	30	7	225	250
r41	8	70	19	40	100	66	58	230	280
r42	47	47	11	50	120	18	37	200	300
r43	29	25	20	0	70	9	5	260	330
r44	75	15	13	20	60	55	28	220	300
r45	12	73	9	0	100	31	69	230	290
r46	32	−2	26	0	100	67	11	200	320
r47	40	38	4	80	160	39	49	200	350
r48	−5	74	22	20	100	29	3	220	290
r49	55	23	15	0	100	−4	76	180	300
r50	28	62	18	0	100	21	25	230	250

Cross-dock Cartesian coordinates: $X_w = 35$, $Y_w = 35$.

Table 2
Vehicle capacity for each example.

Examples	Vehicle Capacity
10R-2V	65
11R-2V	75
12R-3V	65
All other Examples	75

still ranges from 10.6% to 22.3%. Moreover, the optimality gap for the exact model consistently grows with the problem size, thus showing a typical behaviour of NP-hard problems. In contrast, the solution time for the SBF approach looks rather independent of the problem size. The percent deviation of the best solution provided by the sweep-based model from the optimal one remains consistently low as shown in the last column of Table 3. A detailed description of the best sets of pickup and delivery routes for examples 10R-2V, 11R-2V and 15R-3V found with the sweep-based model are shown in Tables 4, 5A and 6, respectively. Such tables include the pickup and delivery tours, the total load picked up or delivered by each truck, the vehicle arrival times at the cross-dock after serving the assigned pickup/delivery nodes, the vehicle ready times for starting delivery duties after unloading the shipment on the cross-dock, and the departure times of the outbound vehicles from that facility. In addition, the pickup and delivery tour costs and the total routing cost are also reported.

Vehicle arrival times at the cross-dock from the pickup routes somewhat differ among them but their values tend to be rather closer as the number of vehicles increases. In contrast, the departure times of outbound trucks are within a narrower range in most examples. Graphical representations of the best solutions for examples 10R-2V, 11R-2V and 15R-3V provided by the SBF are displayed in Figs. 2, 3A and 4, respectively. Pickup and delivery tours present the classical tear-drop shape and no crossing points between edges arise. In Example 10R-2V, requests (r8, r10) are picked up and delivered by vehicle V1 and consequently they are not unloaded at the

Table 3
Best solutions for examples involving 10-to-15 requests using the exact and the sweep-based formulations.

Example	Exact Formulation (EF)			Sweep-based Formulation (SBF)			
	Best solution	Relative gap[a]	CPU[b] (s)	Best solution	Relative gap[a]	CPU (s)	Deviation %
10R-2V	404.13	–	105.3	404.13	–	2.1	–
11R-2V	414.30	–	889.7	422.87	–	35.7	2.06
12R-3V	479.76	0.106	10,800	486.71	–	111.6	1.45
13R-3V	513.94	0.118	10,800	513.78	–	231.9	−0.03
14R-3V	551.65	0.169	10,800	553.24	–	209.3	0.29
15R-3V	589.93	0.223	10,800	583.00	–	56.4	−1.17

[a] Relative gap tolerance = 0.001.
[b] CPU time limit = 10,800 s.

Table 4
The minimum-routing cost solution for Example 10R-2V.

Pickup stage				
Vehicle	Tour	Load Collected	Arrival time	Tour cost
V1	r6–r5–r8–r7–r10	59	101.3	87.0
V2	r1–r9–r3–r4–r2	65	126.8	111.3
Vehicle	Transfer Operations		Ready time	Departure time
V1	Drop-off: r5–r6–r7/Pick-up: r1–r3–r9		118.8	166.8
V2	Drop-off: r1–r3–r9/Pick-up: r5–r6–r7		146.8	164.3
Delivery stage				
Vehicle	Tour	Load Delivered	Arrival time	Tour cost
V1	r8–r10–r3–r9–r1	64	310.5	128.5
V2	r4–r2–r6–r7–r5	60	256.2	77.4
Total routing cost				404.2

Table 5A
The best cost solution for Example 11R-2V using the sweep-based approach.

Pickup stage				
Vehicle	Tour	Load Collected	Arrival time	Tour cost
V1	r10–r7–r11–r8–r5–r6	71	110.0	92.8
V2	r1–r9–r3–r4–r2	65	126.8	111.3
Vehicle	Transfer Operations		Ready time	Departure time
V1	Drop-off: r5–r7–r8–r10–r11/Pick-up: r1–r2–r3–r4–r9		144.5	192.8
V2	Drop-off: r1–r2–r3–r4–r9/Pick-up: r5–r7–r8–r10–r11		159.8	194.3
Delivery stage				
Vehicle	Tour	Load Delivered	Arrival time	Tour cost
V1	r3–r9–r1–r2–r6–r4	68	341.3	131.9
V2	r5–r7–r11–r10–r8	68	297.2	86.9
Total routing cost				422.9

Table 5B
Minimum-routing cost solution for Example 11R-2V using the exact model.

Pickup stage				
Vehicle	Tour	Load Collected	Arrival time	Tour cost
V1	r10–r7–r11–r8–r5–r6	71	110.0	92.8
V2	r1–r9–r3–r4–r2	65	126.8	111.3
Vehicle	Transfer Operations		Release time	Departure time
V1	Drop-off: r7–r10–r11/Pick-up: r4–r2		127.0	153.8
V2	Drop-off: r2–r4/Pick-up: r7–r10–r11		140.3	157.3
Delivery stage				
Vehicle	Tour	Load Delivered	Arrival time	Tour cost
V1	r4–r2–r6–r5–r8	64	251.1	82.1
V2	r1–r9–r3–r10–r11–r7	72	302.8	128.1
Total routing cost				414.3

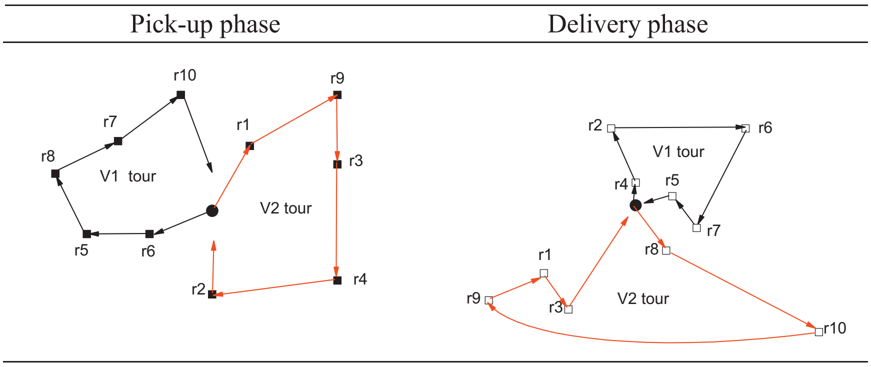

Fig. 2. The minimum-routing cost solution for Example 10R-2V.

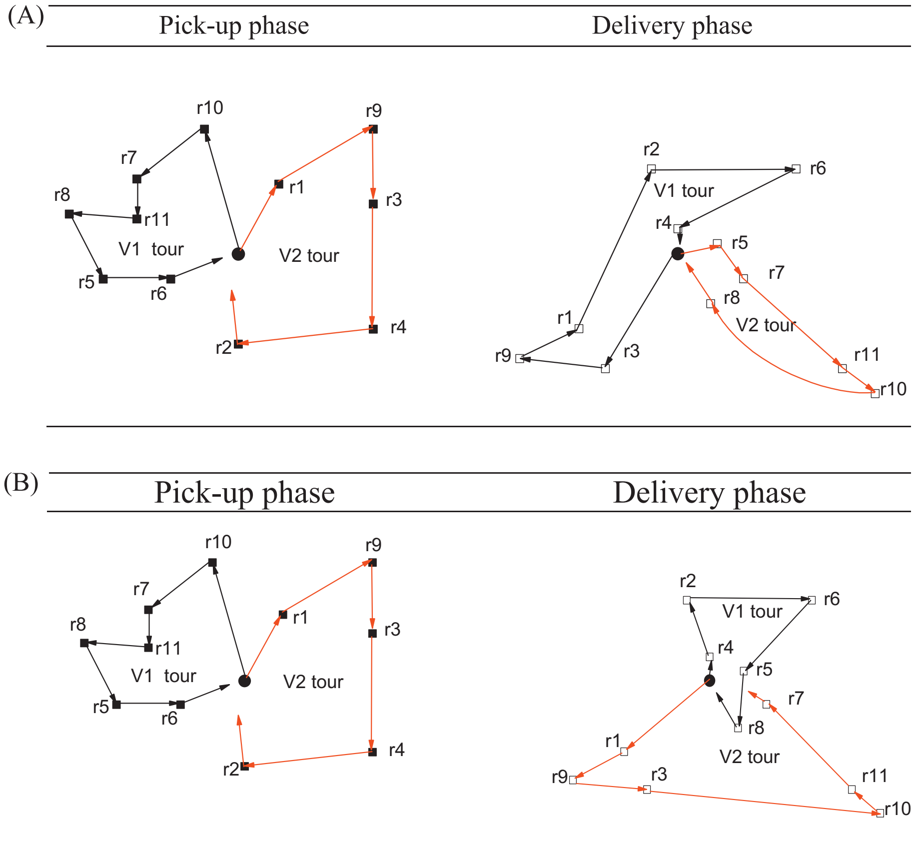

Fig. 3. (A) The best cost solution for Example 11R-2V using the sweep-based approach and (B) the optimal solution to Example 11R-2V.

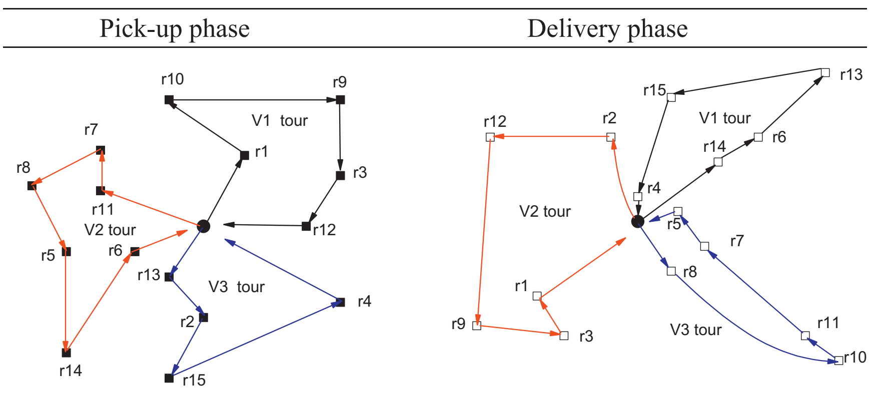

Fig. 4. The best routing-cost solution to Example 15R-3V using the SBF approach.

Table 6
The best routing-cost solution for Example 15R-3V using the SBF.

Pickup stage				
Vehicle	Tour	Load Collected	Arrival time	Tour cost
V1	r11–r7–r8–r5–r14– r6	75	122.2	104.2
V2	r13–r2–r15–r4	57	101.2	87.8
V3	r1–r10–r9–r3–r12	74	114.3	97.0
Vehicle	Transfer Operations		Ready time	Departure time
V1	Drop-off: r5–r7–r8–r11/Pick-up: r4–r13–r15		148.7	174.2
V2	Drop-off: r4–r13–r15/Pick-up: r1–r3–r9–r12		126.7	173.3
	Drop-off: r1–r3–r9–r12/Pick-up: r5–r7–r8–r11		143.8	175.2
Delivery stage				
Vehicle	Tour	Load Delivered	Arrival time	Tour cost
V1	r14–r6–r13–r15–r4	73	281.7	90.3
V2	r2–r12–r9–r3–r1	65	305.6	116.8
V3	r8–r10–r11–r7–r5	68	278.2	86.9
Total routing cost				583.0

Table 7
Best routing cost solutions for examples involving 19-to-50 requests using the sweep-based formulation (SBF).

Example	Routing cost	Relative gap[a]	CPU[b] (s)	Best solution after 1000 s[c]
19R-4V	708.25	–	708.7	708.25
21R-4V	750.22	0.0160	3600	750.22
23R-5V	883.44	0.0249	3600	884.56
25R-6V	893.78	–	1104.9	893.78
28R-6V	995.58	0.0220	3600	995.58
30R-6V	1020.08	0.0163	3600	1020.08
35R-7V	1134.15	0.0571	3600	1140.41
40R-8V	1360.07	0.0337	3600	1360.68
45R-9V	1552.07	0.0164	3600	1574.58
50R-10V	1722.08	–	2011.9	1722.64

[a] Relative gap tolerance = 0.01.
[b] CPU time limit = 3600 s.
[c] Travel cost value.

Table 8
Best routing cost solution for Example 30R-6V using the SBF.

Pickup stage				
Vehicle	Tour	Load Collected	Arrival time	Tour cost
V1	r26–r1–r20–r9–r3–r30	71	97.8	80.6
V2	r10–r19–r7–r8–r11–r18	71	108.9	91.7
V3	r28–r29–r16–r17–r5–r6	68	104.8	72.0
V4	r13–r25–r14–r15–r2	66	100.3	79.8
V5	r24–r21–r23–r22	70	98.3	74.2
V6	r27–r4–r12	60	69.6	56.1
Vehicle	Transfer Operations		Ready time	Departure time
V1	Drop-off: r1–r3–r9–r20–r26–r30/Pick-up: r6–r13–r14–r15–r22		133.8	170.4
V2	Drop-off: r7–r8–r10–r11–r18–r19/Pick-up: r2–r4–r20–r21–r24		144.9	174.4
V3	Drop-off: r5–r6–r16–r17–r28/Pick-up: r12–r23–r26		104.8	165.8
V4	Drop-off: r2–r13–r14–r15–r25/Pick-up: r1–r9–r19–r27		133.8	178.0
V5	Drop-off: r21–r22–r23–r24/Pick-up: r3–r8–r10–r11–r18–r28		133.8	179.9
V6	Drop-off: r4–r12–r27/Pick-up: r5–r7–r16–r17–r25–r30		100.1	179.4
Delivery stage				
Vehicle	Tour	Load Delivered	Arrival time	Tour cost
V1	r14–r6–r13–r22–r15	72	292.4	105.3
V2	r4–r2–r21–r20–r24	58	284.1	95.6
V3	r26–r29–r23–r12	74	258.1	75.5
V4	r27–r9–r19–r1	65	278.2	85.3
V5	r3–r18–r10–r11–r28–r8	69	315.7	119.0
V6	r7–r30–r17–r16–r25–r5	68	281.0	85.0
Total routing cost				1020.1

Table 9
Best routing-cost solution for example 40R-8V using the SBD-based approach.

Pickup stage					
Vehicle	Tour		Load Collected	Arrival time	Tour cost
V1	r30–r31–r3–r12		54	61.7	48.9
V2	r26–r1–r9–r20–r38–r10		72	137.5	120.1
V3	r33–r39–r19–r7–r11		72	96.9	79.9
V4	r6–r5–r17–r8–r18–r34		64	90.4	74.6
V5	r28–r37–r16–r14–r29		74	91.5	74.2
V6	r25–r13–r2–r15–r22–r32		73	97.1	79.5
V7	r27–r23–r21–r24		74	89.9	73.1
V8	r35–r4–r36–r40		57	103.6	90.2
Delivery stage					
Vehicle	Tour	Load delivered	Departure time	Arrival time	Tour cost
V1	r14–r13–r6–r35	60	164.6	261.1	82.4
V2	r15–r36–r22–r4	55	201.9	315.7	100.8
V3	r32–r24–r20–r21–r39–r38–r2	75	201.5	341.6	121.6
V4	r26–r29–r23–r12	74	211.4	303.8	75.5
V5	r27–r9–r19–r1	65	206.9	307.2	85.3
V6	r31–r3–r18–r40–r34–r8	74	171.7	271.3	81.9
V7	r28–r11–r10–r37–r30–r7	65	207.0	316.1	93.1
V8	r16–r17–r33–r25–r5	72	170.6	266.6	79.1
Total routing cost					1360.2

cross-dock. Similarly, requests (r2, r4) remain inside of truck V2 at the cross-dock because V2 is also the assigned outbound vehicle. A similar situation arises in Example 11R-2V where the request r6 is not unloaded from V1 at the cross-dock because V1 is the designated delivery truck.

The optimal solution to Example 11R-2V using the exact formulation is described in Table 5B and Fig. 3B. It is observed that the two vehicle routes overlap each other and, therefore, cannot be found with the sweep-based formulation. The SBF approach only considers solutions with non-overlapping routes, i.e. each vehicle route belongs to a different angular sector. However, the optimal solution to Example 11R-2V can be found through the modified sweep-based model by allowing a maximum overlap $\Delta\theta$ between adjacent routes of 0.8. Convergence to the optimal solution is thus achieved in 1.2 s of CPU time.

At the optimum of Example 11R-2V, requests (r5, r6, r8) are not unloaded on the cross-dock and remain inside V1 while requests (r1, r3, r9) stay into truck V2 after the vehicle arrivals at the cross-dock (see Table 5B). The same pattern is observed in Example 15R-2V where one or more requests remain inside every pickup vehicle after its arrival at the cross-dock.

7.2. *Solving larger examples using the sweep-based approach*

After validating the sweep-based formulation by comparing its results against those provided by the exact model for examples with up to 15 requests, the SBF was applied to solve a set of larger VRPCD problems involving 19–50 customer orders and the minimum transportation cost as the problem goal. Computational results are shown in Table 7. With the exception of examples

Table 10
Best routing-cost solution for problem 50R-10V using the SBF.

Pickup stage					
Vehicle	Tour		Load Collected	Arrival time	Tour cost
V1	r26–r1–r20–r38–r9–r42–r47		71	128.3	110.5
V2	r33–r39–r50–r10		72	92.7	70.5
V3	r45–r48–r41–r19–r7		72	136.6	119.7
V4	r34–r5–r17–r8–r11–r18		73	92.2	74.6
V5	r6–r37–r16–r29–r28		57	72.8	58.9
V6	r13–r43–r14–r25		71	93.2	64.7
V7	r32–r22–r46–r15–r2		68	99.1	83.0
V8	r27–r23–r21–r24		74	89.9	73.1
V9	r4–r44–r36–r49–r35		71	107.3	90.6
V10	r30–r31–r3–r40–r12		68	101.7	85.6
Delivery stage					
Vehicle	Tour	Load delivered	Departure time	Arrival time	Tour cost
V1	r35–r16–r25–r5	67	198.3	286.3	72.6
V2	r14–r6–r13–r41	65	201.5	304.0	87.4
V3	r4–r2–r45–r36–r22–r15–r47	75	202.0	324.3	103.8
V4	r24–r38–r39–r49–r21–r32	74	201.8	352.6	133.1
V5	r29–r23–r20–r12	68	193.3	288.6	79.8
V6	r26–r42–r50–r27	66	197.8	260.7	47.7
V7	r9–r19–r43–r1	63	200.5	301.7	86.5
V8	r40–r48–r18–r3–r31	75	206.6	304.6	80.6
V9	r28–r37–r10–r11–r34–r8	73	180.3	294.1	96.2
V10	r7–r30–r44–r46–r17–r33	71	200.5	320.9	103.2
Total routing cost					1722.1

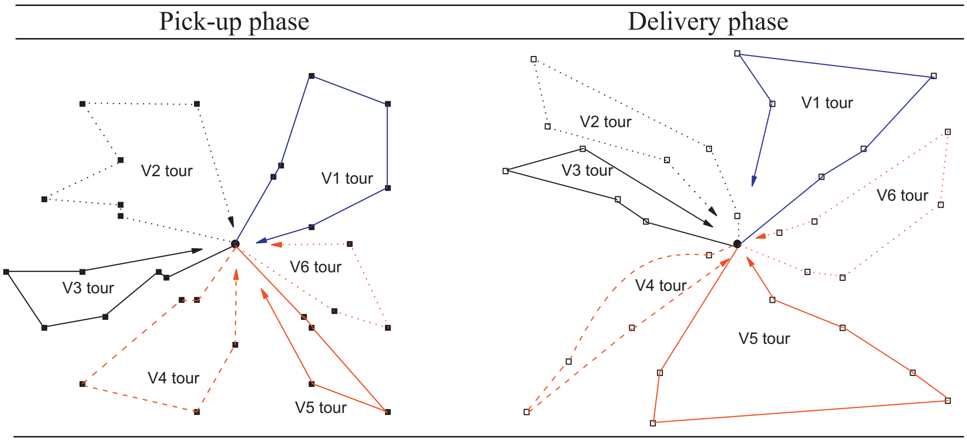

Fig. 5. Best routing-cost solution for example 30R-6V using the SBF-based approach.

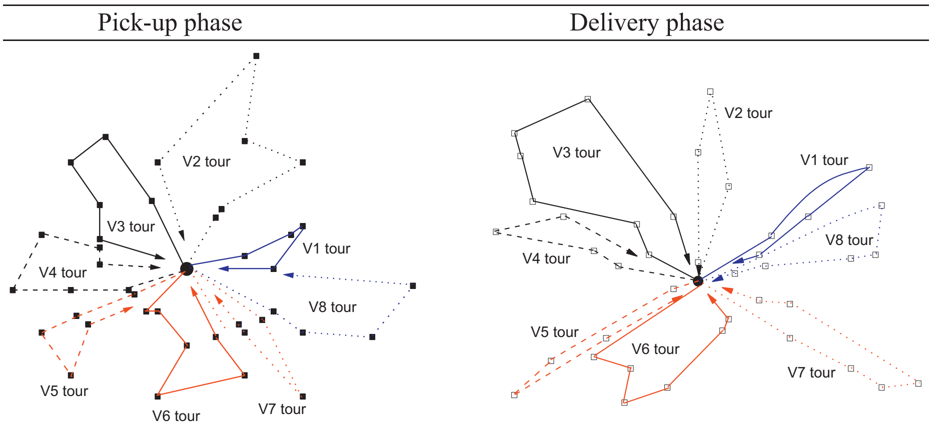

Fig. 6. Best routing-cost solution for example 40R-8V using the SBF-based approach.

19R-4V, 25R-4V and 50R-10V, the optimality has not been reached within the CPU time limit of 3600 s. However, the percent optimality gap is relatively low ranging from 1.6 to 5.7%. Interestingly, it is not observed an exponential increase of the CPU time with the problem size. Detailed descriptions of the best solutions found with the SBF for Examples 30R-6V, 40R-8V and 50R-10V are given in Tables 8–10. Graphical representations of pickup and delivery routes for such examples are depicted in Figs. 5–7. Again, all routes have tear-drop shapes and no crossing points between edges are observed.

Available meta-heuristic techniques for the VRPCD problem (Wen et al., 2009; Liao et al., 2010) based on tabu search algorithms provide good solutions at low computational cost but are not able to prove the solution optimality. Similarly, the sweep-based model does not guarantee optimality but it finds good solutions in reasonable CPU times. The last column of Table 7 reports the best

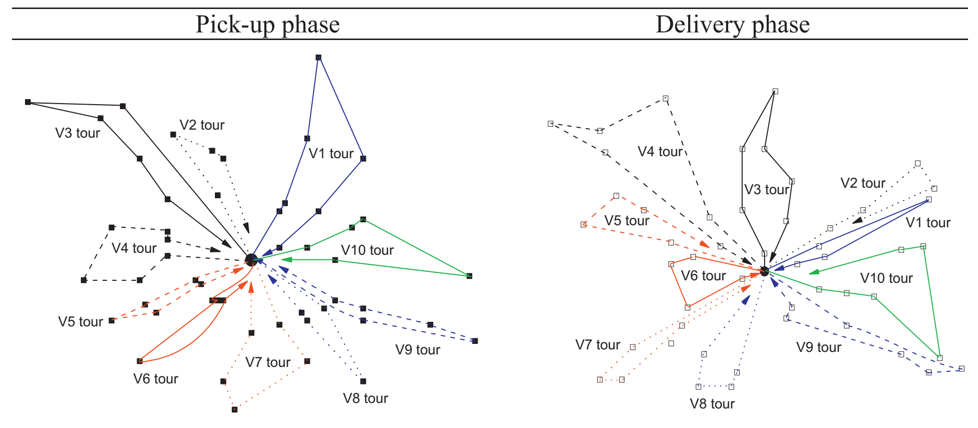

Fig. 7. Best routing-cost solution for example 50R-10V using the SBF.

Table 11
Minimum-makespan solutions for examples with 10R-to-50R using both the sweep-based and the exact formulations.

Example	$\Delta\theta$	SB-model Makespan	Optim.gap[a]	CPU[b] Time (s)	Exact model Makespan	CPU[b] Time (s)	Optim.gap[a]	Deviation %
10R-2V	–	280.87	–	0.8				
	1.50	244.21	–	32.7	244.21	8.7	–	–
11R-2V	–	283.66	–	3.7				
	1.50	249.06	–	65.1	249.06	40.4	–	–
12R-3V	–	211.58	–	5.8	205.70	1743.4	–	2.86
13R-3V	–	236.00	–	8.2				
	0.15	234.86	–	13.7	212.31	2766.7	–	10.72
14R-3V	–	264.03	–	20.8	227.38	3600	0.012	16.12
15R-3V	–	292.17	–	6.8				
	0.70	278.84	–	127.5	249.47	3600	0.146	11.77
19R-4V	–	294.56	–	80.7				
21R-4V	–	311.07	–	56.8				
23R-5V	–	292.88	–	699.7				
25R-5V	–	303.38	–	139.5				
28R-6V	–	280.35	–	483.2				
30R-6V	–	303.21	–	1690.0				
35R-7V	–	279.27	–	690.0				
40R-8V	–	300.67	–	1624.0				
45R-9V	–	324.99	0.132	3600.0				
50R-10V	–	349.07	–	1862.1				

[a] Relative gap tolerance = 0.01.
[b] CPU time limit = 3600 s.

Table 12
Minimum-makespan solution for Example 30R-6V using the SBF.

Pickup stage				
Vehicle	Tour	Load Collected	Arrival time	Tour cost
V1	r30–r3–r9–r20–r1–r26	71	97.8	80.6
V2	r18–r11–r7–r19–r10	62	92.4	77.5
V3	r8–r17– r16–r5–r6–r28	66	101.0	84.8
V4	r13–r15–r14–r29–r25	70	98.5	82.0
V5	r21–r23–r22–r2	65	92.4	77.4
V6	r12–r4–r27–r24	72	74.9	58.4
Vehicle	Transfer Operations		Ready time	Departure time
V1	Drop-off: r1–r3–r9–r20–r26–r30/Pick-up: r6–r13–r14–r16–r25		133.8	171.5
V2	Drop-off: r7–r10–r11–r18–r19/Pick-up: r2–r4–r15–r21–r22–r24		123.9	172.0
V3	Drop-off: r5–r6–r8–r16–r17–r28/Pick-up: r12–r20–r23–r29		134.5	169.0
V4	Drop-off: r13–r14–r15–r25–r29/Pick-up: r9–r19–r26–r27		134.0	170.0
V5	Drop-off: r2–r21–r22–r23/Pick-up: r1–r3–r8–r18–r28		125.4	160.5
V6	Drop-off: r4–r12–r24–r27/Pick-up: r5–r7–r10–r11–r17–r30		111.4	169.4
Delivery stage				
Vehicle	Tour	Load Delivered	Arrival time	Tour cost
V1	r25–r16–r13–r6–r14	73	276.5	87.9
V2	r24–r21–r22–r15–r2–r4	75	303.2	113.2
V3	r12–r20–r23–r29	68	264.3	79.8
V4	r26–r9–r19–r27	70	280.4	94.4
V5	r1–r3–r18–r28–r8	51	266.7	93.5
V6	r7–r30–r11–r10–r17–r5	69	294.2	108.0
Total routing cost				1037.5

solutions found for examples with 19-to-50 customer requests after 1000 CPU seconds through the SBF. They mostly coincide with those discovered within the CPU usage limit of 3600 s. In other words, the proposed formulation can find good solutions in quite acceptable CPU times.

7.3. Minimizing the makespan

An alternative problem goal is the minimization of the makespan. The whole set of VRPCD problem instances previously studied were again solved but this time looking for the set of pickup/delivery routes minimizing the makespan. From the computational results shown in Table 11, it can be inferred that the new objective function is computationally more attractive because all but one of the examples were solved to optimality within the CPU time limit of 3600 s. Even the convergence to the best solution reported for Example 45R-9V was achieved in less than 1000 CPU seconds. Moreover, it is not observed an exponential increase of the solution time with the problem size. In Table 11, true optimal solutions for examples involving up to 15 requests found through the exact model are also given. The last column of Table 11 reports the percent deviation between the results provided by both the sweep-based approach and the exact model. Larger examples were solved using only the sweep-based formulation.

From Table 11, it follows that the sets of pickup and delivery routes minimizing the makespan usually present some overlapping between nearby tours ($\Delta\theta > 0$). This explains why the modified sweep-based formulation leads to better solutions. Moreover, it is observed that the average deviation from the true optimal value is larger than the one found for the minimum

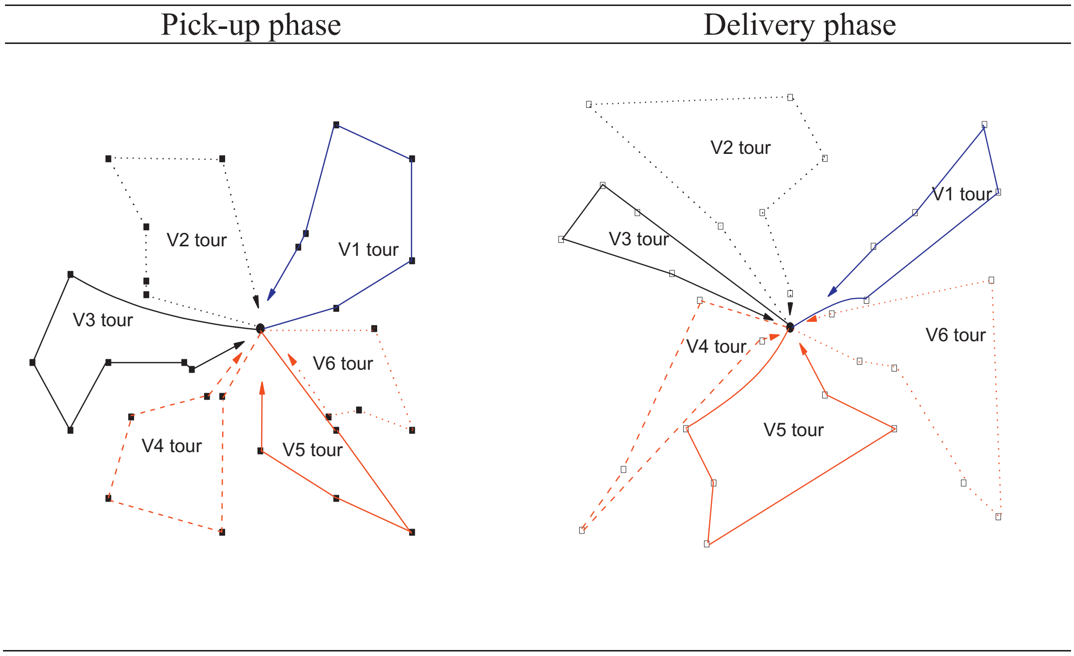

Fig. 8. The best makespan-solution for Example 30R-6V using the SBF approach.

routing-cost target. Interestingly, the computational efficiency of the exact model also improves using the makespan as the problem objective. A detailed description of the best makespan-solution for Example 30R-6V provided by the sweep-based approach is given in Table 12 and Fig. 8.

7.4. Solving VRPCD examples with time windows

When the pickup/delivery nodes of a request should be visited by the designated vehicles within specific time windows, it usually occurs that some overlapping between adjacent routes appears at the optimum. This feature already arises when the minimum makespan is the problem goal. As previously stated, the sweep-based formulation only accounts for solutions with non-overlapping routes. When time windows are considered, it may occur that there is no feasible solution with non-overlapping tours and the proposed sweep-based model will obviously fail. This is the case for examples 10R-2V-TW, 11R-2V-TW, 15R-3V-TW when the time windows given in Table 1 are considered. To discover a feasible or a better solution with the sweep-based approach,

Table 13
Minimum-cost solutions for Examples involving 10-to-50 requests with time windows using the SBF with/without tour overlapping and the exact model.

Example	Sweep-based Approach				Exact Model		
	Max-angular overlap ($\Delta\theta$)	Best routing cost	Relative gap[a]	CPU[b] (s)	Optimum routing cost	CPU[b] (s)	Deviation %
10R-3V-TW	0	445.3	–	5.7			
	1.0	438.1	–	12.5	438.1	21.5	–
11R-3V-TW	0	462.3	–	8.2	462.3	529.3	–
12R-3V-TW	0	551.7	–	4.2			
	1.0	505.2	–	68.0	505.2	103.0	–
13R-3V-TW	0	587.9	–	2.6			
	1.0	534.3	–	28.0	534.3	125.8	–
14R-3V-TW	0	649.7	–	4.8			
	1.0	597.2	–	28.8	589.3	274.6	1.34
15R-4V-TW	0	659.4	–	26.2			
	0.5	628.7	–	192.4	628.7	3600[c]	–
19R-4V-TW	0.3	834.0	–	212.6			
21R-4V-TW	0.5	889.3	–	148.8			
23R-5V-TW	0.2	961.6	–	393.4			
25R-5V-TW	0.3	1068.2	–	1827.2			
28R-6V-TW	0.3	1045.8	–	1633.1			
30R-6V-TW	0	1191.5	–	63.7			
	0.2	1065.1	0.098	3600			
35R-7V-TW	0	1183.2	–	397.8			
	0.1	1135.4	0.084	3600			
40R-8V-TW	0	1474.0	–	1158.4			
	0.05	1394.0	0.018	3600			
45R-9V-TW	0.05	1544.2	0.029	3600			
50R-10V-TW	0	1777.4	–	1183.3			

[a] Relative gap tolerance = 0.01.
[b] CPU time limit = 3600 s.
[c] Relative gap = 0.227.

Table 14
Best routing-cost solution for problem 30R-6V-TW using the SBF approach with tour overlapping.

Pickup stage					
Vehicle	Tour		Load collected	Arrival time	Tour cost
V1	r30–r3–r9–r20		46	118.8	80.4
V2	r26– r1–r10–r7–r19–r11		75	122.0	91.3
V3	r6–r5–r16–r17–r8–r18		71	108-7	83.7
V4	r13–r25–r14–r29–r28		69	130.5	66.9
V5	r2–r15–r22–r23–r21		73	123.0	94.1
V6	r24–r27–r4–r12		72	123.5	58.4
Delivery stage					
Vehicle	Tour	Load delivered	Departure time	Arrival time	Tour cost
V1	r16–r14–r25–r5	73	197.0	292.1	75.5
V2	r6–r13–r22–r15–r4	71	196.0	339.8	105.4
V3	r26– r12–r20–r21–r24–r2	73	207.5	337.5	97.9
V4	r29–r23–r1–r27	72	206.4	328.5	90.3
V5	r9–r19–r18–r3–r8	67	201.2	311.6	101.7
V6	r7–r10–r11–r28–r30–r17	50	194.0	358.5	119.5
Total routing cost					1065.1

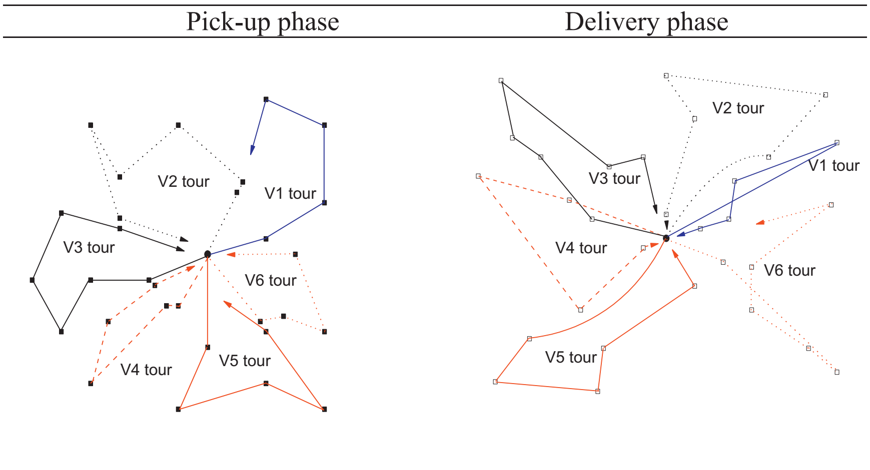

Fig. 9. Best routing-cost solution for example 30R-6V-TW using the SBF with tour overlapping.

there are two types of remedies: (a) increasing the number of available vehicles; (b) keeping the same number of vehicles but increasing the size of the feasible region by also considering solutions with overlapping routes through the modified sweep-based VRPCD formulation. In Examples 10R-2V-TW, 11R-2V-TW and 15R-3V-TW, the number of available vehicles was increased by one to find feasible solutions. Moreover, better feasible solutions for those examples were found by allowing some route overlapping ($\Delta\theta > 0$).

Computational results for the whole set of VRPCD-TW examples are included in Table 13. The selected problem goal is the minimization of the total routing cost. The best solution provided by the SBF admitting tour overlapping is reported just in case it is the only way to find a feasible one or a better set of routes has been discovered. In the case of Examples 19R-4V-TW, 21R-4V-TW, 25R-5V-TW and 45R-9V-TW, it was allowed route overlapping without changing the number of trucks to find good feasible solutions. For other examples, the SBF with overlapping is applied just to get a better set of routes. True optimal solutions using the exact formulation are also reported for examples with up to 15 requests in Table 13. As follows from the last column of Table 13, good agreement between the exact and the sweep-based approaches is achieved by allowing tour overlapping. A detailed description of the best solution found for Example 30R-6V-TW is given in Table 14 and Fig. 9. In this case, some nearby tours overlap each other and crossing points arise on some routes.

8. Conclusions

To address the vehicle routing problem with cross-docking (VRPCD), two different solution approaches have been developed: a rigorous MILP formulation and a sweep-heuristic based MILP model. The latter one was derived from the rigorous representation by including a set of problem constraints that mimics the VRP sweep algorithm. Thus, a more efficient allocation of vehicles to pickup/delivery nodes is achieved although optimality is no longer guaranteed. Both approaches assume that the cross-dock facility has a sufficiently large number of doors so that every truck can immediately start unloading/loading operations after it arrives at the terminal or it becomes ready for delivery duties. Pickup and delivery tasks are carried out by the same fleet of homogeneous vehicles. Moreover, goods collected and delivered by the same vehicle are not unloaded at the cross-dock, and pickup/delivery splits are not allowed.

The sweep-heuristic based approach was first validated by solving a series of small-to-medium size problems and comparing the results obtained with those found using the exact formulation (EF). For these examples, the sweep-based model practically provides near-optimal solutions in a much shorter CPU time. For some examples, it even provides better solutions than the exact formulation within the allowed CPU usage limit. For examples with $N \geq 12$, the exact approach has not converged after a CPU time of 10,800 s and the optimality gap significantly increases with the number of requests. In contrast, the solution time for the SBF looks rather independent of the problem size. In several examples, some requests are picked up and delivered by the same vehicle and consequently they are not unloaded at the cross-dock. After validation, the SBF formulation was applied to larger problem instances involving up to 50 customer requests and the least transportation cost or the minimum makespan as the problem objective. In most cases, convergence to good feasible solutions was achieved within the CPU time limit. Interestingly, the minimum makespan is computationally more attractive as the problem target because all but one of the examples have been solved to optimality. However, the average percent deviation from the true optimal solution is larger compared with the routing-cost objective. Similar to meta-heuristic techniques for the VRPCD problem, the proposed sweep-based approach does not guarantee optimality but it tends to find very good solutions at acceptable computational cost. To illustrate this feature, the best solution found by the sweep-based model after 1000 CPU seconds were also reported. They mostly coincide with those discovered within the CPU usage limit of 3600 s. When the pickup/delivery nodes of a request should be visited within the specified time windows, it usually occurs that there is some overlapping between adjacent tours at the optimum. To handle time window constraints, the modified sweep-based model allowing tour overlapping was applied to find good solutions. At these solutions, some nearby tours overlap each other and crossing points on some routes also arise. Future work will consider a finite number of receiving/shipping dock doors and the allocation/sequencing of vehicles to/at receiving/shipping dock doors, capacity constraints at the cross-dock temporary storage, several commodities and multiple cross-dock platforms.

Acknowledgments

Financial support received from FONCYT-ANPCyT under Grant PICT 01837, from CONICET under Grant PIP-2221, and from Universidad Nacional del Litoral under CAI+D D 66335 is fully appreciated.

References

Bartholdi, J. J., & Gue, K. R. (2004). The best shape for a crossdock. *Transportation Science*, *38*, 235–244.

Boloori Arabani, A. R., Fatemi Ghomi, S. M. T., & Zandieh, M. (2011). Meta-heuristics implementation for scheduling of truck in a cross-docking system with temporary storage. *Expert Systems with Applications*, *38*, 1964–1979.

Boysen, N. (2010). Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research*, *37*, 32–41.

Boysen, N., & Fliedner, M. (2010). Cross dock scheduling: Classification, literature review and research agenda. *Omega*, *38*, 413–422.

Boysen, N., Fliedner, M., & Scholl, A. (2010). Scheduling inbound and outbound trucks at cross docking terminal. *OR Spectrum*, *32*, 135–161.

Chen, P., Guo, Y., Lim, A., & Rodrigues, B. (2006). Multiple crossdocks with inventory and time Windows. *Computers & Operations Research*, *33*, 43–63.

Dondo, R. G., Méndez, C. A., & Cerdá, J. (2009). Managing distribution in supply chain networks. *Industrial & Engineering Chemistry Research*, *48*, 9961–9978.

Dondo, R. G., Méndez, C. A., & Cerdá, J. (2011). The multi-echelon vehicle routing problem with cross docking in supply chain management. *Computers and Chemical Engineering*, *35*, 3002–3024.

Gillett, B., & Miller, L. (1974). A heuristic algorithm for the vehicle dispatch problem. *Operations Research*, *22*, 340–349.

Gumus, M., & Bookbinder, J. H. (2004). Cross-docking and its implications in location-distribution system. *Journal of Business Logistics*, *25*, 199–229.

Jayaraman, V., & Ross, A. (2003). A simulated annealing methodology to distribution network design and management. *European Journal of Operational Research*, *144*, 629–645.

Lee, Y. H., Jung, J. W., & Lee, K. M. (2006). Vehicle routing scheduling for cross-docking in the supply chain. *Computers & Industrial Engineering*, *51*, 247–256.

Li, Z. P., Low, M. Y. H., Shakeri, M., & Lim, Y. G. (2009). Crossdocking planning and scheduling: Problems and algorithms. *SIMTech Technical Reports*, *10*(July–September (3)), 159–167.

Liao, Ch.-J., Lin, Y., & Shih, S. C. (2010). Vehicle routing with cross-docking in the supply chain. *Expert Systems with Applications*, *37*, 6868–6873.

Lim, A., Miao, Z., Rodrigues, B., & Xu, Z. (2005). Transshipment through crossdocks with inventory and time windows. *Naval Research Logistics*, *52*, 724–733.

Miao, Z., Yang, F., Fu, K., & Xu, D. (2012). Transshipment service through crossdocks with both soft and hard time windows. *Annals of Operations Research*, *192*, 21–47.

Ratliff, H. D., Vate, J. V., Zhang, M. (1990). *Network design for load-driven cross-docking systems* (Technical Report). Atlanta: The Logistics Institute, Georgia Institute of Technology.

Ross, A., & Jayaraman, V. (2008). An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design. *Computers and Industrial Engineering*, *55*(1), 64–79.

Sung, C. S., & Song, S. H. (2003). Integrated service network design for a cross-docking supply chain network. *Journal of the Operational Research Society*, *54*, 1283–1295.

Tsui, L., & Chang, C. (1992). An optimal solution to a dock door assignment problem. *Computers & Industrial Engineering*, *23*(1–4), 283–286.

Wen, M., Larsen, J., Clausen, J., Cordeau, J.-F., & Laporte, G. (2009). Vehicle routing with cross-docking. *Journal of the Operational Research Society*, *60*, 1708–1718.

Yu, W., & Egbelu, P. J. (2008). Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research*, *184*, 377–396.