Computers and Chemical Engineering 101 (2017) 259–278

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Solution methods for vehicle-based inventory routing problems

Yachao Dong[a], Christos T. Maravelias[a,*], Jose M. Pinto[b], Arul Sundaramoorthy[c]

[a] Department of Chemical and Biological Engineering, University of Wisconsin – Madison, 1415 Engineering Drive, Madison, WI 53706, United States
[b] Praxair Inc., Business and Supply Chain Optimization R&D, Danbury, CT 06810, United States
[c] Praxair Inc., Business and Supply Chain Optimization R&D, Tonawanda, NY 14150, United States

ARTICLE INFO

Article history:
Received 8 July 2016
Received in revised form 18 January 2017
Accepted 19 February 2017
Available online 3 March 2017

Keywords:
Vendor managed inventory
Mixed-integer programming
Network reduction algorithm
Decomposition method

ABSTRACT

A novel method for solving vehicle-based inventory routing problems (IRPs) under realistic constraints is presented. First, we propose a preprocessing algorithm that reduces the problem size by eliminating customers and network arcs that are irrelevant for the current horizon. Second, we develop a decomposition method that divides the problem into two subproblems. The upper level subproblem considers a simplified vehicle routing problem to minimize the distribution cost while satisfying minimum demands, which are calculated based on consumption rate, initial inventory and safety stock. In the lower level, a detailed schedule with drivers is acquired using a continuous-time MILP model, by adopting the routes selected from the upper level. Finally, an iterative approach based on the upper and lower levels is presented, including the addition of different types of integer cuts and parameter updates. Different options of implementing this iterative approach are discussed, and computational results are presented.

1. Introduction

Vendor managed inventory (VMI) policies are increasingly being adopted in many sectors. In a VMI policy, the vendor no longer receives orders from customers as in the traditional approach, but actively monitors the inventory levels of customers, and decides when and how much to serve. Such policies can be beneficial to both the vendor and customers. On the vendor's side, a substantial amount of savings can be achieved, since customers can be combined in routes more freely and distributions are scheduled more efficiently (Disney et al., 2003). On the customers' side, inventory levels are maintained by the vendor within their preferred bounds, as the VMI agreement states, which leads to cost savings on inventory management. To make the distribution decisions under VMI policy, including route selection, delivery amount and delivery time, the decision maker (vendor) needs to solve an inventory routing problem (IRP). The vendor and customers are normally viewed as different business entities, but the same idea may be applied to internal distributions of the same entity. In other words, IRP generally appears in distribution networks between up-stream and down-stream nodes of the supply chain (SC), when the vehicle routing and delivery scheduling are addressed simultaneously with inventory management decisions.

There are many types of IRPs studied in the literature, which can be categorized in terms of inventory policy, fleet type, and network structure. When serving customers, several inventory policies can be adopted: bringing the customer inventory level to its maximum capacity, to a predefined target level or to any level as long as the inventory bounds are respected (Coelho and Laporte, 2015). The fleet can be homogeneous or heterogeneous in terms of capacities (Savelsbergh and Song, 2007; Hewitt et al., 2013). The network structure is either single-vendor and multi-customer, or multi-vendor and multi-customer. The former mostly appears in vehicle-based transportation, while the latter arises in maritime settings, which are referred to as maritime IRPs (Adulyasak et al., 2015; Papageorgiou et al., 2014a). In general, IRP can include either single-product or multi-product distribution. In the latter case, dedicated or undedicated compartments can be required (Siswanto et al., 2011; Jetlund and Karimi, 2004; Al-Khayyal and Hwang, 2007). Furthermore, IRP arises in different industrial sectors, such as petrochemicals, commodity chemicals, industrial gases, grocery and department stores (Singh et al., 2015; Gaur and Fisher, 2004; Christiansen et al., 2011; Shen et al., 2011). Recently, the integration of IRP and production has also been studied (Zhang et al., 2017).

To address different types of IRPs, different mixed-integer linear programming (MILP) models, as well as solution methods, have been proposed. Archetti et al. (2007) modeled IRPs under VMI with order-up-to level policy or maximum level policy; while Avella et al. (2015) considered a reformulation for similar problems with constant demand over time. For maritime IRP, Song and

* Corresponding author.
E-mail address: maravelias@wisc.edu (C.T. Maravelias).

http://dx.doi.org/10.1016/j.compchemeng.2017.02.036
0098-1354/

Notation

Indices/sets

$i \in \mathbf{I}$ trucks
$j \in \mathbf{J}$ SC nodes, including plant P
$k \in \mathbf{K}$ drivers
$l \in \mathbf{L}$ segments
$m \in \mathbf{M}_j$ access windows of customer j
$n \in \mathbf{N}$ time slots
$q \in \mathbf{Q}$ piecewise linear approximation points
$r \in \mathbf{R}$ routes
$s \in \mathbf{S}$ iterations

Subsets

$\mathbf{A} \subseteq (\mathbf{J} \times \mathbf{J})$ arcs
$\mathbf{A}_l/\mathbf{A}_r$ arcs included in segment l/route r
$\mathbf{A}^{RP}_{r,j}$ arcs traveled before j in route r
$\mathbf{I}_l$ trucks that can carry out segment l
$\mathbf{I}^E_s$ trucks that are assigned to infeasible routes in OptnE
$\mathbf{J}^C$ customers
$\mathbf{J}^A/\mathbf{J}^O$ anticipatable/order-only customers
$\mathbf{J}^{first}$ customers required to be visited first in a route
$\mathbf{J}_l$ SC nodes visited in segment l
$\mathbf{J}^{start}_l/\mathbf{J}^{end}_l$ start/end SC node of segment l
$\mathbf{J}_r$ customers visited in route r
$\mathbf{J}^T/\mathbf{J}^B$ trigger/balance customers
$\mathbf{J}^R_j$ customers in the region of j
$\mathbf{L}^S$ single-route segments
$\mathbf{L}^1/\mathbf{L}^2$ first/second segments of long routes
$\mathbf{L}_j$ segments visiting customer j
$\mathbf{L}^{next}_l$ the second segment, following segment l, in a route
$\mathbf{L}_r$ segments related to route r
$\mathbf{N}^I/\mathbf{N}^J_j/\mathbf{N}^K$ slots of trucks/customer j/drivers
$\mathbf{R}_l$ routes related to segment l
$\mathbf{R}_j$ routes visiting customer j
$\mathbf{R}^G_{s,i}/\mathbf{R}^E_{s,i}/\mathbf{R}^R_s$ infeasible route combinations (for different types of integer cuts)

Parameters

β_j fixed loading or delivering time at SC node j
$\gamma^D/\gamma^R/\gamma^W$ driving/resting/working cost
γ^V/γ^X delivery/unused capacity cost
ε termination criterion
$\zeta^L_j/\zeta^U_j/\zeta^S_j$ minimum/maximum/safety level of anticipatable customer j
$\overline{\zeta}_{j,q}$ projected inventory level at point q of customer j without deliveries
η planning horizon
θ^W/θ^D maximum daily working/driving time
$\overline{\lambda}_{j,q}$ time at point q of customer j
ξ_i capacity of truck i
ρ_j constant consumption rate of anticipatable customer j
$\rho^T_j(t)$ consumption rate of anticipatable customer j at time t
$\sigma^{AS}_{j,m}/\sigma^{AE}_{j,m}$ start/end time of access window m of customer j
$\sigma^{OS}_j/\sigma^{OE}_j$ start/end time of order window of customer j
$\tau_{j,j'}$ traversal time of arc (j,j') including loading or delivering time at j
$\tau o_{j,j'}$ travel time of arc (j,j')
φ_j order amount of order-only customer j
$\varphi^{CI}/\varphi^{CO}$ check-in/check-out time
ψ minimum resting time
$\omega_{i,j}$ variable time for a unit material delivery from truck i to customer j
$L0^A_j$ initial inventory of anticipatable customer j

Calculated parameters

$\alpha\tau_{r,j}$ earliest possible visiting time to customer j via route r
γ^R_r cost of route r
ϑ_j fixed working time at SC node j
μ_r number of times that route r is selected in VR solution
$\sigma^{\mathrm{MIN}}_j/\sigma^{\mathrm{MAX}}_j$ minimum/maximum demand in the planning horizon of customer j
$\tau^W_r/\tau^D_r/\tau^R_r$ working/driving/routing time of route r
$\tau x_{i,r}$ updated extra working time of route r by truck i
$\omega\tau_j$ time when the projected inventory of customer j goes below lower bound

Binary variables in VR

$Z_{i,r}$ 1 if and only if truck i uses route r

Continuous non-negative variables in VR

$F^{RX}_{i,r}$ unused capacity of truck i when carrying out route r
$F^R_{i,r,j}$ delivery amount from truck i to customer j in route r
O^{VR} objective in VR

Binary variables in SP

$X_{i,n,k,n',l}$ 1 if and only if truck-slot (i,n) is matched with driver-slot (k,n') to carry out segment l
$X^I_{i,n}$ 1 if and only if slot n of truck i is used
$X^K_{k,n}$ 1 if and only if slot n of driver k is used
$X^{IL}_{i,l}$ 1 if and only if truck i carries out segment l
$Y_{l,j,n}$ 1 if and only if segment l visits customer j on slot n
$W_{i,n,k,n',l,j,m}$ 1 if and only if truck-slot (i,n) is matched with driver-slot (k,n') to carry out segment l, in which customer j is visited during window m

SOS2 variables in SP

$P^S_{j,n,q}/P^E_{j,n,q}$ SOS2 over index q representing start/end time on slot n of customer j

Continuous non-negative variables in SP

$F_{i,n,k,n',l,j}$ delivery amount to customer j at truck-slot (i,n) and driver-slot (k,n') on segment l
$F^{LJ}_{l,j}$ delivery amount on segment l to customer j
$F^{JN}_{j,n}$ delivery amount to customer j at slot n
$\hat{F}^L_{j,n}/\hat{F}^U_{j,n}$ inventory lower/upper bound violation for customer j at slot n
$F^{SX}_{i,l}$ unused capacity for truck i on segment l
$L^S_{j,n}/L^E_{j,n}$ projected inventory level at the start/end of slot n of customer j (which can be negative)
O^{SP} objective in SP

$S_{i,n,k,n',l,j}/E_{i,n,k,n',l,j}$ start/end time to visit SC node j using truck-slot (i,n) and driver-slot (k,n') on segment l
$\hat{S}_{i,n,k,n',l,j}/\hat{E}_{i,n,k,n',l,j}$ start/end time violation
$S^I_{i,n}/E^I_{i,n}$ start/end time of slot n of truck i
$S^K_{k,n}/E^K_{k,n}$ start/end time of slot n of driver k
S^L_l/E^L_l start/end time of segment l
$S^{lJ}_{l,j}/E^{lJ}_{l,j}$ start/end time to visit SC node j on segment l
$S^{JN}_{j,n}/E^{JN}_{j,n}$ start/end time to visit SC node j on slot n

Furman (2013) proposed a model using discrete-time approach, while both continuous- and discrete-time models were studied by Jiang and Grossmann (2015). Moreover, different heuristic methods have been developed, including methods based on valid inequalities (Persson and Göthe-Lundgren, 2005; Song and Furman, 2013), column generation (Grønhaug et al., 2010; Hewitt et al., 2013; Desaulniers et al., 2016), Lagrangian decomposition (Yu et al., 2006; Shen et al., 2011), genetic algorithms (Aziz and Moin, 2007) and other decomposition-based algorithms (Jetlund and Karimi, 2004; Campbell and Savelsbergh, 2004). IRP has also been solved in a cyclic approach (Raa, 2015), as well as using a fuzzy approach with multiple objective functions (Niakan and Rahimi, 2015). More details can be found in review papers (Baita et al., 1998; Moin and Salhi, 2007; Andersson et al., 2010; Coelho et al., 2014; Papageorgiou et al., 2014b).

Despite the research in the field, real-world vehicle-based IRPs cannot be solved efficiently, primarily due to numerous practical constraints that have to be satisfied. These constraints include variable consumption rates, customer access windows, and driver constraints. Most importantly, drivers are a critical resource subject to strict rules. These rules set limits on driving, working and resting time, according to the Department of Transportation requirements or, more strictly, the company's requirements. Considering the drivers' rules, models and solution methods have been developed for driver scheduling and vehicle routing problems (Goel, 2009; Goel, 2012; Rancourt et al., 2013). A MILP model for IRP that addresses all these constraints has been proposed (Dong et al., 2014), but it becomes intractable for large instances. Accordingly, the goal of this paper is to address this challenge. Specifically, we propose solution methods to address the computational difficulties of solving vehicle-based IRPs. While we use an industrial gas SC as an example, the methods are general; i.e., they can be applied to vehicle-based IRPs in other industries.

The article is structured as follows. In Section 2, we describe the supply chain under VMI, provide a detailed problem statement, and summarize the solution methods. In Section 3, we present a "dynamic" network preprocessing algorithm that reduces the problem size by eliminating irrelevant SC nodes and network arcs for the current horizon. In Section 4, an upper level vehicle routing (VR) model is presented, which deals with the simplified vehicle routing problem to minimize the distribution cost while satisfying minimum customer demand. In Section 5, a lower level scheduling problem (SP) model is proposed, which yields a detailed schedule for each truck and driver, using the routes selected in the upper level. In Section 6, we present an iterative approach that integrates the two subproblems. In Section 7, different instances are presented. Throughout the paper, we use lowercase italic letters for indices, uppercase bold letters for sets, and uppercase italic letters for variables. Lowercase Greek letters are used for parameters, except for a few calculated parameters denoted by combinations of Greek letters. Subsets are denoted by the letter for the superset and a superscript; e.g., $\mathbf{J}^A$ (anticipatable customers) is a subset of $\mathbf{J}$ (all supply chain nodes). Superscripts are also used to differentiate variables and parameters.

2. Problem and method overview

We first discuss VMI policies and the corresponding IRP; then, we present a detailed problem statement; finally, we summarize the solution methods proposed in the paper. The IRP addressed here is based on an industrial gases SC, but the problem in other sectors is very similar.

2.1. Supply chain under VMI policy

A distribution network consists of plants, customers, storage facilities, trucks (each associated with a trailer) and drivers. Under VMI, most customer inventories are managed by the vendor, i.e., the vendor installs storage facilities in customer locations with proper sizes and manages their replenishments. The vendor proactively monitors the inventories of customers in real time, by installing communicating units termed Remote Telemetry Units. The vendor can then decide when and how much to deliver to each customer to satisfy demand.

A fleet of trailers of various capacities are employed in a certain region. The product is carried on a variety of tanker-trailers, and it is transferred to the storage tank at each customer through different routes. In this article, a *route* means an ordered set of arcs, $\{a_1, a_2, \ldots, a_n\}$, in which the end node of a_i and the start node of a_{i+1} are the same, and the plant is the start node of a_1 and the end node of a_n. Routes can be broadly classified as: single-customer routes and multi-customer routes.

In a *single-customer route*, a trailer departs from the plant, delivers all or most of the product to a customer, and then directly returns to the plant. These routes are typically for customers with a storage tank of sufficient capacity to hold the entire volume of the trailer. Occasionally, there are also emergency deliveries made to customers with smaller capacities, in order to prevent stockouts.

In a *multi-customer route*, a trailer departs from the plant, delivers the product to multiple customers, and then returns to the plant. Customers with smaller storage tanks are typically served on such routes.

Long-term decisions involve the number of tanks to install in each customer location and the size of each tank (You et al., 2011). Other long-term decisions include when and how to install new tanks at customer locations, as well as when and how to upgrade and downgrade existing tanks. Short-term distribution decisions include which customers to deliver to each day, when and how much to deliver, how to combine deliveries into routes, how to fit routes into drivers' schedules, and which truck or trailer to assign to each route. In this article, we consider the short-term decisions.

2.2. Problem statement

The problem is represented in terms of the following sets:

(a) $i \in \mathbf{I}$: trucks;
(b) $k \in \mathbf{K}$: drivers;
(c) $j \in \mathbf{J}$: SC nodes, including a central plant P, and a subset $\mathbf{J}^C$, denoting customers.

Each truck i is associated with a trailer tank of capacity ξ_i. For each driver, a maximum daily working/driving time should be respected, i.e., a driver cannot work/drive more than θ^W/θ^D hours per day. Also, a driver cannot work again until he has remained *off duty* for at least ψ consecutive hours. For a route that cannot be finished within the working/driving time limits, the driver can take a

ψ-hour rest on the road; we will refer to this type of route as a *long route*.

The customers are classified as either *anticipatable* customers, $j \in \mathbf{J}^A$ (i.e., customers whose inventory are forecasted and maintained by the vendor), or *order-only* customers, $j \in \mathbf{J}^O$. Also, some customers should be visited first in a route, denoted by $\mathbf{J}^{first}$. Each customer may have multiple access windows in the horizon: for a window, $m \in \mathbf{M}_j$, during which customer j can receive products, we know its start/end time, $\sigma^{AS}_{j,m}/\sigma^{AE}_{j,m}$. If traveling from j to j' is infeasible or too expensive, the arc (j,j') is removed from the set of arcs in the SC network, $\mathbf{A} \subseteq \mathbf{J} \times \mathbf{J}$. The travel time along an arc (j,j') is $\tau o_{j,j'}$. The product loading time at the plant ($j = P$) and the delivering time at the customers ($j \in \mathbf{J}^C$), both denoted by β_j, are fixed; i.e., they do not depend on the loading/delivering amount. Under this assumption, the traversal time ($\tau_{j,j'}$) of each arc can be calculated to include the travel time and the fixed loading/delivering time at the start SC node, i.e., $\tau_{j,j'} = \beta_j + \tau o_{j,j'}$. In Section 5, we discuss the case in which the loading/delivering time is not fixed.

An anticipatable customer may have variable consumption rate (e.g., high during the day and low or zero during the night). The consumption profile in the planning horizon is assumed to be an input, calculated from demand forecasts prior to optimization. For each anticipatable customer $j \in \mathbf{J}^A$, we are also given the capacity, ζ^U_j, of the tank and the minimum inventory level, ζ^L_j. At any time, the inventory level is required to be within these two bounds.

We assume that an order-only customer has at most one order placed in the current planning horizon, though this assumption can be easily relaxed by introducing a set of orders, $o \in \mathbf{O}_j$, placed by $j \in \mathbf{J}^O$. An order from customer j is described by the amount, φ_j, as well as the start and end time, σ^{OS}_j and σ^{OE}_j, within which the order has to be satisfied.

The objective is to find the optimal routes, delivery amounts, schedules, and resource allocations (drivers, trucks), to minimize the distribution cost. We assume that there is only one central plant, in which the products are always available. No loss during transportations and deliveries is considered, though it can be easily modeled. It is also assumed that there is only one product, as different products are often distributed by different trailers and scheduled independently. In practice, drivers are shared among products, but here we assume that drivers are also dedicated to products.

2.3. Solution strategy

The proposed solution strategy includes three components, described in Sections 3–5. First, we reduce the distribution network *dynamically*, using the current inventory levels, demand rates and geographical information of the customers. Specifically, we eliminate nodes (customers) and arcs that can be neglected in the current planning horizon. Then, we adopt a decomposition method, which includes an upper level vehicle routing subproblem and a lower level scheduling subproblem.

After the network reduction, we generate the routes to visit customers. In the upper level subproblem, we solve a vehicle routing model; this model selects the routes to visit customers and decides which truck to carry out each selected route. Based on the decisions in the upper level subproblem, we solve a detailed lower-level scheduling model to determine the driver-truck parings to carry out each route and the delivery times and amounts for each customer. Since the upper level does not consider all the constraints in IRP (i.e., it is a relaxation), the route-selection and truck-route-paring decisions might lead to an infeasible or sub-optimal lower level model. To address this, we iterate between the upper and lower level subproblems, using integer cuts to obtain different upper-level solutions. The iterative approach, with different options, is described in Section 6. A simplified flowchart of the solution approach is shown in Fig. 1.

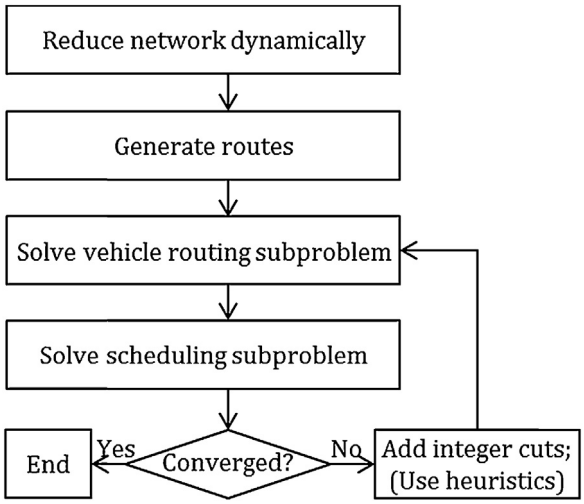

Fig. 1. Outline of the solution strategy.

3. Dynamic network reduction

One major difficulty in solving IRP stems from the large size of the distribution network, which leads to computationally intractable MILP models. However, when solving a specific instance at a specific time point, not all customers and customer-customer arcs have to be considered. Thus, we propose a dynamic network reduction method that returns a sub-network which contains the relevant SC nodes and arcs for the current planning horizon.

Since we address a detailed IRP whose parameters are updated in real time, its horizon is relatively short. Thus, only a small proportion of customers are required to be visited within the horizon. These customers are called *"trigger" customers*, denoted by $\mathbf{J}^T$. Furthermore, some other customers should also be included, so that truck capacities are fully utilized, and the distribution cost in the long run is minimized. These customers are referred to as *"balance" customers*, denoted by $\mathbf{J}^B$. A balance customer should be "close" to the arc connecting the plant to a trigger customer, and also have some vacant capacity to receive more product. In addition, arcs connecting the customers that are not included in the sub-network are eliminated. Due to long distance or road construction, some arcs which are very unlikely to be used are also eliminated.

3.1. Customer selection

In the first step, we identify the trigger and balance customers to be included in the current sub-network.

Trigger customers include the order-only customers that have pending orders within the horizon, as well as anticipatable customers that are expected to run out of product if no deliveries take place. Let $\rho^T_j(t)$ denote the time-varying consumption rate of customer j, and $L0^A_j$ denote its initial inventory. The *minimum* and *maximum demand* for each customer can be calculated as follows:

$$\sigma^{\mathrm{MIN}}_j = \begin{cases} \max\left(0, \zeta^S_j + \int_0^{\eta} \rho^T_j(t)\,dt - L0^A_j\right) & \text{if } j \in \mathbf{J}^A \\ \varphi_j & \text{if } j \in \mathbf{J}^O \end{cases} \quad (1)$$

$$\sigma^{\mathrm{MAX}}_j = \begin{cases} \zeta^U_j + \int_0^{\eta} \rho^T_j(t)\,dt - L0^A_j & \text{if } j \in \mathbf{J}^A \\ \varphi_j & \text{if } j \in \mathbf{J}^O \end{cases} \quad (2)$$

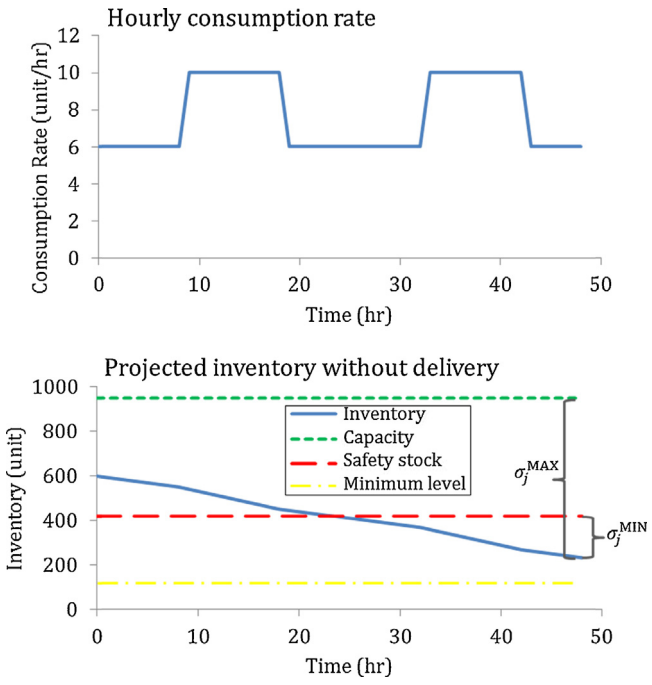

Fig. 2. The procedure of determining trigger customers.

The minimum demand of an anticipatable customer is calculated based on its consumption rate, initial inventory and safety stock level, while the maximum demand is calculated from the consumption rate, initial inventory and tank capacity. For an order-only customer, both the minimum and maximum demands are equal to the order amount. If the minimum demand is greater than zero, then this customer is included in the set of trigger customers, i.e., $\mathbf{J}^T = \left\{ j \middle| \sigma_j^{\text{MIN}} > 0 \right\}$. This idea is illustrated in Fig. 2.

If safety stock levels are not given, they can be calculated using the equation below (Eppen and Martin, 1988),

$$\zeta_j^S = \max\left\{ a \cdot \zeta_j^U, \zeta_j^L + \overline{\tau}_{Pj} \cdot \overline{\rho}_j + b \cdot \sqrt{\overline{\tau}_{Pj} \cdot \delta^2(\rho_j) + \overline{\rho}_j^2 \cdot \delta^2(\tau_{Pj})} \right\} \quad (3)$$

This tentative safety stock is a maximum of two terms. The first term requires safety level to be greater than the minimum reserve stock level, where a is the minimum reserve level percentage. The second term consists of three parts. The first part is a lower bound of stock level ζ_j^L, while the second and third parts are based on statistical data on travel time and consumption rate. Here, both the travel time, τ_{Pj}, from the plant to this customer and consumption rate, ρ_j, are treated as random variables: $\overline{\tau}_{Pj}/\overline{\rho}_j$ are their mean values, and $\delta^2(\tau_{Pj})/\delta^2(\rho_j)$ are their variances. As a time-invariant safety stock is preferred, consumption rate of each customer is treated as a random variable with a time-invariant distribution. With these assumptions, the second part $\overline{\tau}_{Pj} \cdot \overline{\rho}_j$ is the average demand during the travel time from the plant to the customer; the third part is a buffering term for the uncertainty of travel time and consumption rate. The vendor can specify a service level (i.e., the percentage of cases that the buffering inventory will be sufficient), and parameter b in Eq. (3) is associated with this service level. More specifically, 1 minus the specified service level is the upper tail of a standard normal distribution at b.

To fully utilize the capacities of trucks, *balance customers* are included into the current SC sub-network. They should have capacity to receive more product, and be in the vicinity of the line extending from the plant to a trigger customer so that distribution cost will not increase substantially. Thus, two types of criteria

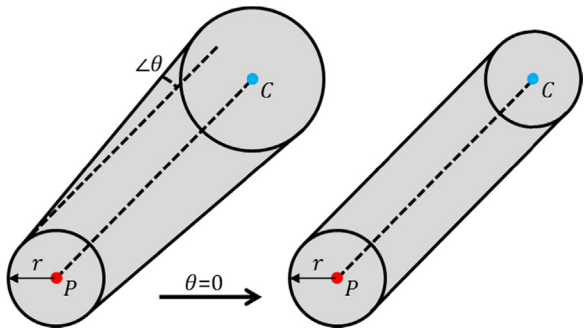

Fig. 3. Illustration of the trigger customer region. C is the trigger customer, and P is the plant.

are used simultaneously to identify the set of potential balance customers, based on the geographical locations and inventory levels.

In terms of geography, a balance customer is required to be in one of the trigger customer regions. The region of customer j should be close to the radial line that extends from the plant to this customer, and it can be defined based on longitude and latitude information (see Fig. 3 and Appendix B). The adjustable parameters defining this region are the angle θ, and the radius r. When $\theta = 0$, the shape becomes a stadium. We use $\mathbf{J}_j^R$ to denote the set of customers that are in the region of customer j.

Balance customers should also require a delivery in the near future. To quantify this, we introduce a parameter T_j, defined by the decision maker. A customer j will be included as a balance customer, only if its current inventory level is less than the summation of (i) consumption in the planning horizon, (ii) the consumption in T_j days following the current horizon, and (iii) its safety stock. The bigger T_j is defined, the more likely customer j will be included as a balance customer. We present two options to define T_j. In option A, customers are set into manually determined regions, and customers in each region have the same T_j; the closer a region is to the plant, the smaller T_j will be, because it can be visited more easily (see Fig. 4a). In option B, T_j is defined based on customer density around j. The number of customers within a disk centered at customer j can be calculated. If this number is larger, customer j is located in a "denser" region, and thus has a higher probability to be included as a balance customer. Thus, to avoid including j too frequently, T_j should have a smaller value. Following this reasoning, T_j in option B is defined as follows,

$$T_j = \max\left\{ \overline{T} - \left\lfloor \frac{C_j/r^2}{\overline{C}/\overline{r}^2} \right\rfloor, 1 \right\} \quad (4)$$

where $\overline{T}$ is the user-defined largest possible T_j, $\overline{r}$ is the maximum distance between any customer and the plant, $\overline{C}$ is the number of customer in the network, r is a user-defined neighbor distance (typically, r can be 80 miles, or the average distance a truck can travel in 2 h), C_j is the number of other customers within the disk of radius r around customer j. With T_j defined in Eq. (4), which is illustrated in Fig. 4b, customers in different density regions have about the same probability of being included as balance customers. To consider both the plant-customer distance and customer density, we can use the average value, or any other affine combinations, of T_j defined in options A and B.

To consider both geographical and inventory criteria, the set of balance customers is defined as follows,

$$\mathbf{J}^B = \left\{ j' \middle| j' \in \cup_{j \in \mathbf{J}^T} \mathbf{J}_j^R \text{ and } LO_{j'}^A - \int_0^{\eta + 24T_{j'}} \rho_{j'}^T(t)\, dt < \zeta_{j'}^S \right\} \quad (5)$$

When a trigger customer j does not lead to the inclusion of another customer in $\mathbf{J}^B$, inventory criterion is relaxed, and the customer j'

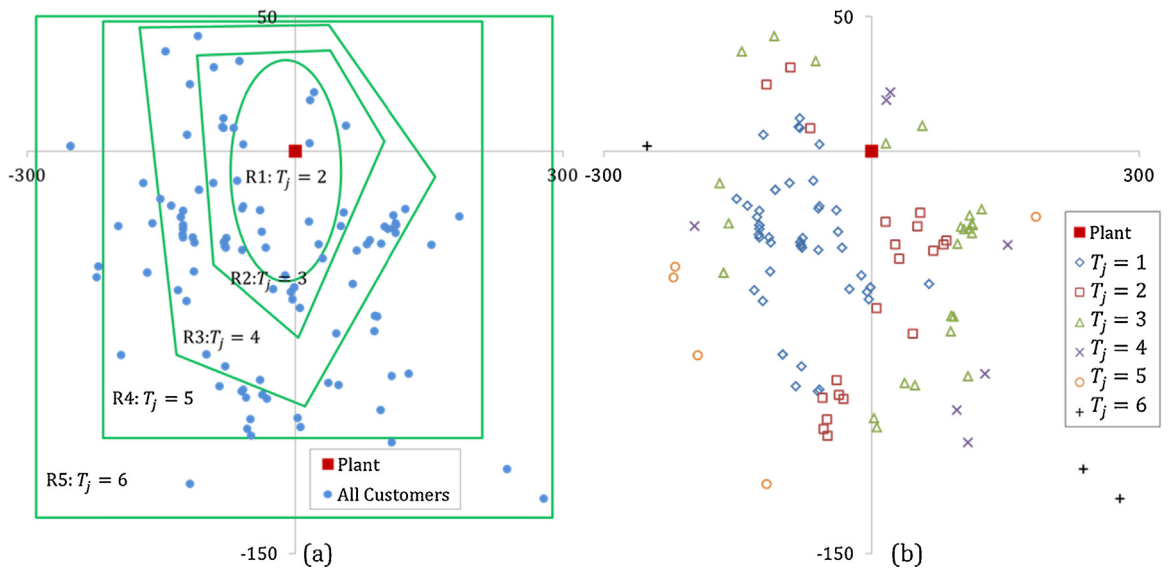

Fig. 4. Illustration of different T_j definition in inventory level criterion, with both axes in unit of miles. (a) Option A to consider plant-customer distance, in which customers are divided into regions R1–R5. (b) Option B to consider customer density.

that is within the trigger customer region and has the greatest $\sigma_{j'}^{MAX}$ is included as a balance customer for j. By doing this, we can ensure that enough balance customers are included after preprocessing so that the truck capacities are fully utilized.

3.2. Network arc elimination

The arcs in the original network are kept in the sub-network, except for the following 4 cases. First, arcs with at least one SC node not in the sub-network are eliminated. Second, a customer-customer arc with very large distance, which is unlikely to be included in the optimal schedule, is eliminated: the following inequality is used to identify these arcs,

$$\tau o_{j,j'} \geq \max[c \cdot \theta^D, d \cdot (\tau o_{j,P} + \tau o_{j',P})] \quad (6)$$

where $\tau o_{j,j'}$ is the travel time along this arc; $\tau o_{j,P}$ and $\tau o_{j',P}$ are the travel time between the customers and the plant; c and d are user-defined parameters. Inequality (6) requires that the travel time from j to j' is greater than both (i) a percentage of the maximum daily driving time and (ii) a percentage of the travel time of $j \rightarrow P \rightarrow j'$. Typically, c and d are selected between 0.7 and 1. Third, if both ends of an arc are balance customers, and they are not in the same trigger customer region, this arc is eliminated. Fourth, optionally, a neighbor list from history data can be used to remove arcs: based on previous routing information, the arcs that have never been used will not appear in the sub-network. The preprocessing algorithm is presented in Appendix A.

3.3. Example

The customer set shown in Fig. 4 is used as an example. The planning horizon is 2 days. Parameter T_j is based on Fig. 4a, and the trigger customer region is defined using option A ($\theta = 10^\circ$ and $r = 10$ miles). The preprocessing algorithm identifies 18 trigger customers, and 14 balance customers (see Fig. 5). The number of customers drops from 111 to 32, and the number of directed arcs drops from 6067 to 485.

4. Vehicle routing subproblem

The upper level subproblem considers the selected customers (both trigger and balance customers) after the dynamic network reduction. Before building the upper level model, which corresponds to a modified vehicle routing (VR) problem, routes ($r \in \mathbf{R}$) for the selected customers are generated, and the corresponding time and cost parameters for each route are calculated. We note that column generation has been adopted to speed up the VR solution process (Grønhaug et al., 2010; Bard and Nananukul, 2010; Persson and Göthe-Lundgren, 2005). However, column generation is not considered here, because the number of generated routes is relatively small, and the resulting VR model can be solved rather fast.

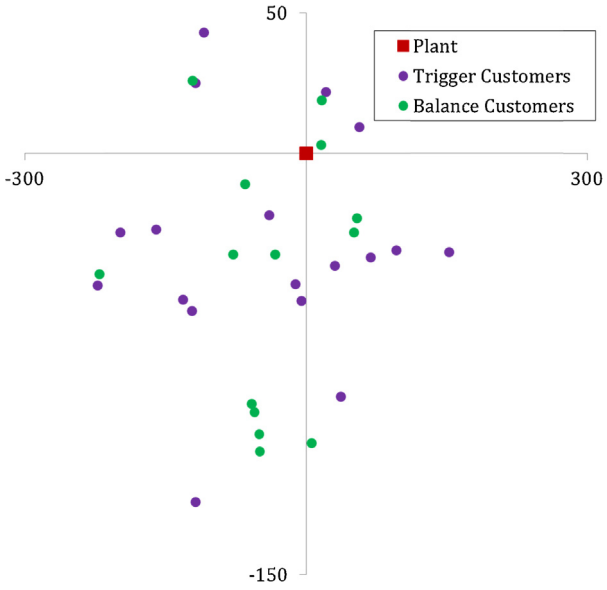

Fig. 5. SC nodes in the distribution network after dynamic network reduction.

4.1. Route generation

In a route, the customers and the sequence in which they are visited are specified. We use $\mathbf{A}_r$ to denote the arcs of a route r, $\mathbf{J}_r$ to denote the set of customers visited in route r, and $\mathbf{R}_j$ to denote the set of routes serving customer j. The following parameters are introduced for each route:

(a) τ_r^D: driving time, based on travel time $\tau o_{j,j'}$.
(b) τ_r^W: working time, based on traversal time $\tau_{j,j'}$ (including loading and delivering), plus possible waiting time due to access window constraints.
(c) τ_r^R: routing time, which is working time plus resting time ψ, if the maximum driving/working time is violated; otherwise, $\tau_r^R = \tau_r^W$.
(d) γ_r^R: routing cost, based on driving time ($\$\gamma^D$/hour), working time ($\$\gamma^W$/hour), number of deliveries ($\$\gamma^V$/delivery), and whether a rest is included in the route ($\$\gamma^R$/rest).

These parameters are calculated as follows,

$$\tau_r^D = \sum_{(j,j')\in \mathbf{A}_r} \tau o_{j,j'} \tag{7}$$

$$\tau_r^W = \sum_{(j,j')\in \mathbf{A}_r} \tau_{j,j'} + \sum_{(j,j')\in \mathbf{A}_r : j,j' \neq P} \max\left(0, \min_m \sigma_{j',m}^{AS} - \max_m \sigma_{j,m}^{AE} - \tau o_{j,j'}\right) \tag{8}$$

$$\tau_r^R = \begin{cases} \tau_r^W & \text{if } \tau_r^D \le \theta^D \text{ and } \tau_r^W \le \theta^W \\ \tau_r^W + \psi & \text{otherwise} \end{cases} \tag{9}$$

$$\gamma_r^R = \begin{cases} \gamma^D \cdot \tau_r^D + \gamma^W \cdot \tau_r^W + \gamma^V \cdot |\mathbf{J}_r| & \text{if } \tau_r^D \le \theta^D \text{ and } \tau_r^W \le \theta^W \\ \gamma^D \cdot \tau_r^D + \gamma^W \cdot \tau_r^W + \gamma^V \cdot |\mathbf{J}_r| + \gamma^R & \text{otherwise} \end{cases} \tag{10}$$

Each route in the generated route set $\mathbf{R}$ should satisfy the following criteria:

(a) The route should contain no more than *cmax* customers; i.e., $|\mathbf{J}_r| \le cmax$. Because of the limited capacities of trucks, it is very unlikely that more than 3 customers are included in one single route in the cases we studied, thus we choose *cmax* to be 3, but it can be generalized depending on the characteristics of a specific SC.
(b) The arcs of the route should be in the valid arc set; i.e., if $(j,j') \in \mathbf{A}_r$, then $(j,j') \in \mathbf{A}$. For example, the 3-customer route, $j \to j' \to j''$, is included in $\mathbf{R}$, only if both arcs (j,j') and (j',j'') are included in the sub-network after dynamic network reduction.
(c) There should be no obvious time conflicts on the access windows of customers; i.e., if $(j,j') \in \mathbf{A}_r$ and $j,j' \neq P$, then $\max_m \sigma_{j',m}^{AE} \ge \min_m \sigma_{j,m}^{AS} + \tau_{j,j'}$. For example, the 2-customer route, $j \to j'$ is included in $\mathbf{R}$, only if the earliest arriving time at customer j' after visiting j is sooner than the end time of the last window of j'.
(d) Based on distance, a truck should be able to arrive at the customer before the end time of its last access window; i.e., if $j \in \mathbf{J}_r$, then $\max_m \sigma_{j,m}^{AE} \ge \sum_{(j',j'')\in \mathbf{A}_{r,j}^{RP}} \tau_{j',j''}$, where $\mathbf{A}_{r,j}^{RP}$ denotes all the arcs in route r before visiting customer j.
(e) A customer in $\mathbf{J}^{first}$ should be visited first in a route; i.e., if $j \in \mathbf{J}_r \cap \mathbf{J}^{first}$, then $(P,j) \in \mathbf{A}_r$.
(f) The first customer visited in a route should be either a trigger customer or in set $\mathbf{J}^{first}$; i.e., if $(P,j) \in \mathbf{A}_r$, then $j \in \mathbf{J}^{first} \cup \mathbf{J}^T$. This requirement is to ensure that the demands of trigger customers are met in face of uncertainties.

We also include some optional criteria based on heuristic rules. By doing this, some routes that are very unlikely to appear in the optimal schedule are excluded:

(g) The total time of a route should not be so long that more than one rest is required; i.e., $\tau_r^W \le 2\theta^W$ and $\tau_r^D \le 2\theta^D$.
(h) If the route includes more than two customers, the route should not include any customer whose demand can be satisfied by one visit of a truck, and at the same time, whose capacity allows for a full truck load; i.e., if $|\mathbf{J}_r| > 2$ and $j \in \mathbf{J}_r$, then $\sigma_j^{\text{MIN}} > \min_i \xi_i$ or $\sigma_j^{\text{MAX}} < \max_i \xi_i$. This is because such a customer can be served more efficiently using a 1-customer or 2-customer route.

The algorithm to generate routes is given in Appendix A. The route generation process is effective in filtering a large proportion of the infeasible routes; based on the instances studied in Section 7.2, more than 80% of routes (which include up to 3 customers) are excluded.

4.2. Vehicle routing model

We present a modified capacitated VR model. Comparing to the standard VR model (Gounaris et al., 2013), we add constraints on the upper bounds of customer demands and truck routing time. The drivers are not modeled here. First, we introduce the following variables:

(a) $Z_{i,r} \in \{0,1\}$ is one if truck i is assigned to route r.
(b) $F_{i,r,j}^R \ge 0$: delivery amount from truck i to customer j using route r.
(c) $F_{i,r}^{RX} \ge 0$: unused capacity (full truck load minus deliveries) of truck i when carrying out route r.
(d) O^{VR}: objective value of VR, corresponding to total distribution (routing) cost with penalized unused capacity.

The VR model is formulated as follows,

$$\min O^{VR} = \sum_{i,r} (\gamma_r^R Z_{i,r} + \gamma^X F_{i,r}^{RX}) \tag{11}$$

$$\sum_{j\in \mathbf{J}_r} F_{i,r,j}^R + F_{i,r}^{RX} = \xi_i Z_{i,r}, \quad \forall i, r \tag{12}$$

$$F_{i,r,j}^R \le (\zeta_j^U - \zeta_j^L) Z_{i,r}, \quad \forall i, r, \quad j \in \mathbf{J}^A \cap \mathbf{J}_r \tag{13}$$

$$\sigma_j^{\text{MIN}} \le \sum_{i,r\in \mathbf{R}_j} F_{i,r,j}^R \le \sigma_j^{\text{MAX}}, \quad \forall j \in \mathbf{J}^C \tag{14}$$

$$\sum_r \tau_r^R Z_{i,r} \le \eta, \quad \forall i \tag{15}$$

The objective function (11) accounts for the routing cost, and a penalty term for unused truck capacity (γ^X per unit of material). Constraints (12) enforce the truck capacity, and fix the delivery amounts to zero if route r is not used by truck i. Constraints (13) enforce that each delivery cannot exceed the difference between the maximum and minimum inventory levels, while constraints (14) enforce demand satisfaction for each customer. Constraints (15) state that the total routing time of a truck should be less than the horizon length.

Two additional sets of constraints can be added to reduce either the computational cost for the VR model, or the number of iterations between the upper and lower level subproblems. The first set of constraints is defined as follows,

$$\sum_{i,r\in \mathbf{R}_j : \alpha\tau_{r,j} \le \omega\tau_j} Z_{i,r} \ge 1, \quad \forall j \in \mathbf{J}^A \tag{16}$$

where $\omega\tau_j$ denotes the time when the projected inventory of customer j (without delivery) goes below its lower bound (defined in

Eq. (17)), and $\alpha\tau_{r,j}$ denotes the earliest possible time to visit j on route r (defined in Eq. (18)). Thus, constraints (16) enforce that at least one route whose $\alpha\tau_{r,j}$ is less than $\omega\tau_j$ should be selected to prevent j from running out of product.

$$\omega\tau_j = \min_t \left\{ t \left| LO_j^A - \int_0^t \rho_{j,t'}^T \, dt' \le \zeta_j^L \right. \right\} \tag{17}$$

$$\alpha\tau_{r,j} = \sum_{(j',j'')\in \mathbf{A}_{r,j}^{RP}} \tau_{j',j''} \tag{18}$$

The second constraints enforce that if customer j has demand which cannot be fulfilled by a single truck, a full truck delivery should be used at least once,

$$\sum_{i,r\in \mathbf{R}_j:|\mathbf{J}_r|=1} Z_{i,r} \ge 1, \quad \forall j \in \mathbf{J}^A : \sigma_j^{\mathrm{MIN}} \ge \max_i \xi_i \tag{19}$$

where $r \in \mathbf{R}_j{:}|\mathbf{J}_r| = 1$ is the single-customer route visiting j. Note that constraints (19) may cut off the optimal solution, in some rare cases, of the finite horizon problem; however, in the long run, customers with large demand should be served by full truck deliveries (see Section 2.1).

5. Scheduling subproblem

From the upper level VR solution, the routes are selected, and the truck-route pairings are determined. Based on these decisions, we consider a scheduling problem (SP) using a continuous representation of time.

5.1. Segment generation

First, plant node P is replaced by two SC nodes: Ps, Pe, standing for plant-start and plant-end. To model the resting on the road, we introduce a set of segments, $l \in \mathbf{L}$. There are three types of segments:

(a) $l \in \mathbf{L}^S$: a route that can be finished without driver resting, starting at Ps, and ending at Pe.
(b) $l \in \mathbf{L}^1$: the first segment of a long route, starting at Ps, and ending at a customer.
(c) $l \in \mathbf{L}^2$: the second segment of a long route, starting at the next SC node after the first segment of this route, and ending at Pe.

Throughout the paper, we use these two terms, route and segment, with slightly different meanings. A route is an ordered set of arcs starting from the plant, visiting several customers, and finally coming back to the plant. A segment is an ordered set of arcs that can be finished without driver resting, and it can start or end at a customer. We divide a long route in which a driver needs to rest on the road into two segments. From the end of the first segment, l, to the start of the second, l', the driver travels from the end SC node of l to the start SC node of l', and takes a rest. If segment l is the entire route r ($l \in \mathbf{L}^S$), or part of it ($l \in \mathbf{L}^1 \cup \mathbf{L}^2$), segment l and route r are called *related*. We generate all related segments of each route selected in VR, including all ways to divide a long route, as illustrated in Fig. 6.

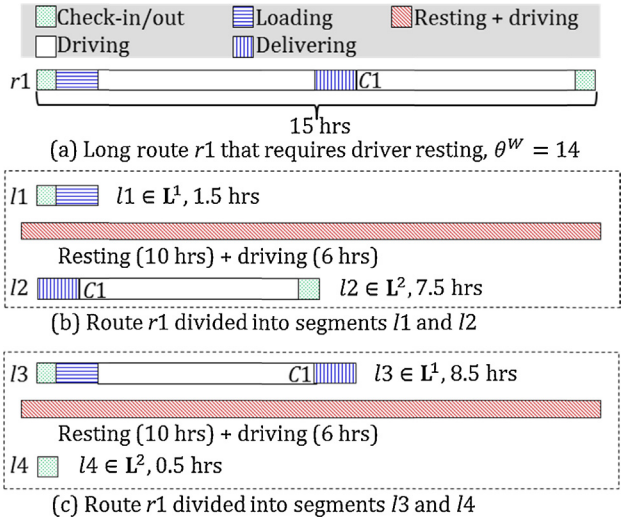

Fig. 6. All ways to break long routes into segments are considered.

Second, sets $\mathbf{R}$, $\mathbf{J}$, $\mathbf{J}^C$, $\mathbf{J}^A$, $\mathbf{J}^O$ are updated, so that only the routes and the customers selected in the solution of VR are included. Index slot $n \in \mathbf{N} = \{1, \ldots, maxN\}$ is introduced, to model different routes of the same truck, different segments assigned to the same driver, and different visits to the same customer (see Fig. 7a). Specifically, the following sets are defined:

(a) $\boldsymbol{N}^l = \{1, \ldots, N^{maxi}\} \subseteq \mathbf{N}$: route slots for trucks, where N^{maxi} is the maximum number of routes that a truck is assigned to in the VR solution, i.e., $N^{maxi} = \max_i \left(\sum_r Z_{i,r} \right)$.

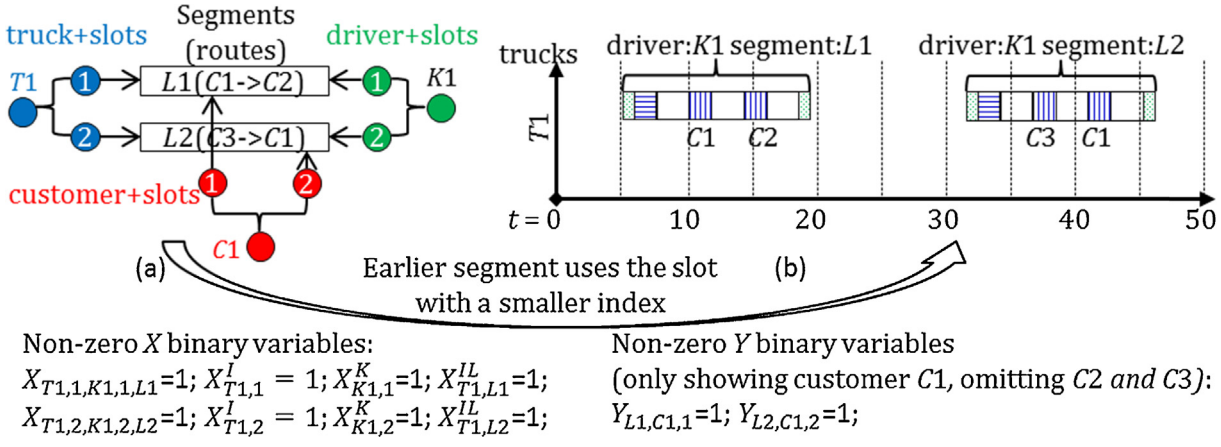

Fig. 7. Illustration of slots and binary variables; two routes/segments are assigned to the same truck ($T1$) and driver ($K1$), and one customer ($C1$) appears in both routes.

(b) $\mathbf{N}_j^J = \left\{1, \ldots, N_j^{\max}\right\} \subseteq \mathbf{N}$: customer slots, where $N_j^{\max}$ is the times that customer j is visited in the VR solution, i.e., $N_j^{\max} = \sum_{i,r \in \mathbf{R}_j} Z_{i,r}$.

(c) $\mathbf{N}^K = \left\{1, \ldots, N^{maxk}\right\} \subseteq \mathbf{N}$: segment slots for drivers, where N^{maxk} is the maximum number of segments a driver can have, which is the maximum of two terms as follows,

$$N^{maxk} = \max\left\{\left\lceil \frac{\sum_{i,r} Z_{i,r} + \sum_{i,r:\tau_r^D > \theta^D \text{ or } \tau_r^W > \theta^W} Z_{i,r}}{|\mathbf{K}|} \right\rceil, \max_i \left(\sum_r Z_{i,r} + \sum_{r:\tau_r^D > \theta^D \text{ or } \tau_r^W > \theta^W} Z_{i,r} \right) \right\}$$

In the first term, the numerator is the number of segments to be carried out based on the VR solution, where $\sum_{i,r:\tau_r^D > \theta^D \text{ or } \tau_r^W > \theta^W} Z_{i,r}$ is added as a correction for long routes with driver resting; the denominator is the cardinality of the driver set. The second term denotes the maximum number of segments a truck can be assigned to; this ensures enough driver slots if a truck is assigned to a single driver.

Third, we define the following subsets:

(a) $\mathbf{A}_l \subseteq \mathbf{A}$: arcs included in segment l.
(b) $\mathbf{I}_l \subseteq \mathbf{I}$: trucks that can carry out segment l.
(c) $\mathbf{J}_l \subseteq \mathbf{J}$: SC nodes visited in segment l.
(d) $\mathbf{J}_l^{start}/\mathbf{J}_l^{end} \subseteq \mathbf{J}$: first/last SC node in segment l.
(e) $\mathbf{L}_j \subseteq \mathbf{L}$: segments visiting customer j.
(f) $\mathbf{L}_l^{next} \subseteq \mathbf{L}$: the second segment in a long route after segment $l \in \mathbf{L}^1$.
(g) $\mathbf{L}_r \subseteq \mathbf{L}$: segments related to route r.
(h) $\mathbf{R}_l \subseteq \mathbf{R}$: route related to segment l.

We also calculate the following parameters:

(i) $\mu_r \in \mathbb{Z}$: the times route r is selected in the current VR solution.
(j) $\vartheta_j \in \mathbb{R}$: the fixed working time at SC node j. Specifically, for a customer $j \in \mathbf{J}^C$, it is the fixed delivering time (β_j); for plant-start Ps, the checking-in time plus loading time ($\beta_P + \varphi^{CI}$); for plant-end Pe, the checking-out time (φ^{CO}).

5.2. Variables

The following binary variables are introduced:

(a) $X_{i,n}^I = 1$ if slot n of truck i is used.
(b) $X_{k,n}^K = 1$ if slot n of driver k is used.
(c) $X_{i,l}^{IL} = 1$ if truck i carries out segment l.
(d) $X_{i,n,k,n',l} = 1$ if slot n of truck i is matched with slot n' of driver k to carry out segment l.
(e) $Y_{l,j,n} = 1$ if the visit of segment l is assigned to customer j on slot n.
(f) $W_{i,n,k,n',l,j,m} = 1$ if slot n of truck i is matched with slot n' of driver k to carry out segment l, and customer j is visited on its window m in this segment.
(g) $R_{k,n} = 1$ if slot n of driver k is started after a rest.

The main binary variable is $X_{i,n,k,n',l}$, which represents the segment assignments to trucks and drivers. Variables $X_{i,n}^I X_{k,n}^K X_{i,l}^{IL}$, as aggregated versions of $X_{i,n,k,n',l}$, are introduced to break symmetry and accommodate time constraints, for truck usage, driver usage, and truck-segment pairing respectively (see Fig. 7b, where an earlier segment is assigned to the slot with a smaller index of trucks, drivers and customers). Variable $Y_{l,j,n}$ is used in inventory constraints, while $W_{i,n,k,n',l,j,m}$ and $R_{k,n}$ are used for access window constraints and time limit constraints respectively.

The following continuous non-negative variables are used to model time:

(a) $S_{i,n}^I/E_{i,n}^I$: start/end time of slot n of truck i.
(b) $S_{k,n}^K/E_{k,n}^K$: start/end time of slot n of driver k.
(c) S_l^L/E_l^L: start/end time of segment l.
(d) $S_{l,j}^{LJ}/E_{l,j}^{LJ}$: start/end time of the visit on segment l to SC node j.
(e) $S_{i,n,k,n',l,j}/E_{i,n,k,n',l,j}$: start/end time of visit to SC node j using slot n of truck i and slot n' of driver k on segment l.
(f) $S_{j,n}^{JN}/E_{j,n}^{JN}$: start/end time of visit to customer j on slot n.

The main time variables are $S_{i,n,k,n',l,j}/E_{i,n,k,n',l,j}$. Variables $S_{j,n}^{JN}/E_{j,n}^{JN}$ are introduced for inventory constraints. The remaining time variables, as aggregated versions of $S_{i,n,k,n',l,j}/E_{i,n,k,n',l,j}$, are introduced to express the constraints for different time grids (trucks, drivers, segments and customers).

Finally, the following continuous non-negative variables are used to model material flows,

(a) $F_{l,j}^{LJ}$: delivery amount on segment l to customer j.
(b) $F_{i,n,k,n',l,j}$: delivery amount to customer j using slot n of truck i and slot n' of driver k on segment l.
(c) $F_{j,n}^{JN}$: delivery amount to customer j on slot n.
(d) $F_{i,l}^{SX}$: unused capacity for truck i on segment l.

The main material flow variable is $F_{i,n,k,n',l,j}$, and $F_{l,j}^{LJ}$ is an aggregated version of it. Variable $F_{j,n}^{JN}$ is used for inventory constraints, while $F_{i,l}^{SX}$ is introduced to penalize unused truck capacity.

5.3. Segment assignment constraints

Segments are assigned to different trucks and drivers as follows,

$$\sum_{k,n' \in \mathbf{N}^K, l \notin \mathbf{L}^2} X_{i,n,k,n',l} = X_{i,n}^I \quad \forall i, n \in \mathbf{N}^I \tag{20}$$

$$X_{i,n}^I \geq X_{i,n+1}^I \quad \forall i, n \in \mathbf{N}^I \tag{21}$$

$$\sum_{i,n \in \mathbf{N}^I, l} X_{i,n,k,n',l} = X_{k,n'}^K \quad \forall k, n' \in \mathbf{N}^K \tag{22}$$

$$X_{k,n}^K \geq X_{k,n+1}^K \quad \forall k, n \in \mathbf{N}^K \tag{23}$$

Constraints (20) define the truck aggregated variable $X_{i,n}^I$, while constraints (21) are used for symmetry breaking. Constraints (22) define the driver aggregated variable $X_{k,n}^K$, and constraints (23) break the symmetry in the same way as constraints (21). Note that the summation in constraints (20) excludes the second segment of long routes, $\mathbf{L}^2$, while constraints (22) do not, because the slots of trucks correspond to routes, which can be represented by the first segment for a long route, while the slots of drivers correspond to segments, which can facilitate the driver constraints in Section 5.4.

$$\sum_{n \in \mathbf{N}_i^I, k, n' \in \mathbf{N}^K} X_{i,n,k,n',l} = X_{i,l}^{IL} \quad \forall i, l \tag{24}$$

$$\sum_{i,l \in \mathbf{L}_r \setminus \mathbf{L}^2} X_{i,l}^{IL} = \mu_r \quad \forall r \tag{25}$$

$$X_{i,n,k,n',l} = X_{i,n,k,n'+1,l'} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l \in \mathbf{L}^1, l' \in \mathbf{L}_l^{next} \tag{26}$$

Constraints (24) define the truck-segment aggregated variable $X_{i,l}^{IL}$, while constraints (25) require that the segments which are related to route r, but not a second segment of a long route ($\mathbf{L}^2$), should be carried out as many times as route r is used in the VR solution. Constraints (26) enforce that if the first segment of a long route is assigned to truck-slot (i, n) and driver-slot (k, n'), the second segment of it should be assigned to the same truck (slot n for routes) and driver (slot $n'+1$ for segments). We fix $X_{i,n,k,n',l}$ to zero, if truck i is not in the set of trucks that can carry out segment l ($i \notin \mathbf{I}_l$).

5.4. Time constraints

We constrain the variables of start and end time to respect the visiting sequence and the working and resting time limits. Note that by the definition of segments, the driving time of each segment is given, so the driving time limits are inherently satisfied and not written explicitly.

$$S_{i,n,k,n',l,j} \le \eta \cdot X_{i,n,k,n',l} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \tag{27}$$

$$E_{i,n,k,n',l,j} \le \eta \cdot X_{i,n,k,n',l} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \tag{28}$$

$$S_{i,n}^I = \sum_{k,n' \in \mathbf{N}^K, l \notin \mathbf{L}^2, j \in \mathbf{J}_l^{start}} S_{i,n,k,n',l,j} \quad \forall i, n \in \mathbf{N}^I \tag{29}$$

$$E_{i,n}^I = \sum_{k,n' \in \mathbf{N}^K, l \notin \mathbf{L}^1, j \in \mathbf{J}_l^{end}} E_{i,n,k,n',l,j} \quad \forall i, n \in \mathbf{N}^I \tag{30}$$

$$E_{i,n}^I \le S_{i,n+1}^I + \eta \cdot (1 - X_{i,n+1}^I) \quad \forall i, n \in \mathbf{N}^I \tag{31}$$

Constraints (27)/(28) enforce the start/end time of visiting a SC node, $S_{i,n,k,n',l,j}/E_{i,n,k,n',l,j}$, are zero if the corresponding assignment variable $X_{i,n,k,n',l}$ is zero. Constraints (29)/(30) define the truck start/end time variables $S_{i,n}^I/E_{i,n}^I$, while constraints (31) state that slot $n+1$ of truck i cannot start before slot n of the same truck is finished.

$$S_{k,n'}^K = \sum_{i,n \in \mathbf{N}^I, j \in \mathbf{J}_l^{start}} S_{i,n,k,n',l,j} \quad \forall k, n' \in \mathbf{N}^K \tag{32}$$

$$E_{k,n'}^K = \sum_{i,n \in \mathbf{N}^I, j \in \mathbf{J}_l^{end}} E_{i,n,k,n',l,j} \quad \forall k, n' \in \mathbf{N}^K \tag{33}$$

Constraints (32)/(33) define the driver start/end time variables $S_{k,n}^K/E_{k,n}^K$. (The difference between them and constraints (29), (30) is due to the same reason as for $X_{k,n}^K$ and $X_{i,n}^I$, described in Section 5.3). In practice, a driver may be available only before/after a certain time and for a period smaller than θ^D due to weekly driving limits. These constraints can be easily added using variables $S_{k,n}^K/E_{k,n}^K$.

$$R_{k,n} \le X_{k,n}^K \quad \forall k, n \in \mathbf{N}^K \tag{34}$$

$$S_{k,n+1}^K - E_{k,n}^K \ge \psi \cdot R_{k,n+1} - \eta \cdot (1 - X_{k,n+1}^K) \quad \forall k, n \in \mathbf{N}^K \backslash \{N^{maxk}\} \tag{35}$$

$$E_{k,n+1}^K - S_{k,n}^K \le \theta^W + \eta \cdot R_{k,n+1} \quad \forall k, n \in \mathbf{N}^K \backslash \{N^{maxk}\} \tag{36}$$

Constraints (34)–(36) express restrictions on the working and resting time of drivers. Constraints (34) require $R_{k,n}$ to be zero if $X_{k,n}^K$ is zero. Constraints (35) enforce that if a driver starts its $n+1$ segment (slot) without resting ($R_{k,n+1} = 0$ and $X_{k,n+1}^K = 1$), then $S_{k,n+1}^K \ge E_{k,n}^K$; otherwise, if this segment is started after resting ($R_{k,n+1} = 1$ and $X_{k,n+1}^K = 1$), then $S_{k,n+1}^K \ge E_{k,n}^K + \psi$. Constraints (36) require that if segment $n+1$ is started without resting ($R_{k,n+1} = 0$), then the difference of the end time of segment $n+1$ and the start time of segment of n should be less than the working time limit θ^W.

$$S_{k,n-1}^K + 2\theta^W + \psi \ge E_{k,n+1}^K - \eta \cdot \left(R_{k,n} + 1 - \sum_{i,n' \in \mathbf{N}^I, l \in \mathbf{L}^1} X_{i,n',k,n,l} \right) \quad \forall k, n \in \mathbf{N}^K \backslash \{1, N^{maxk}\} \tag{37}$$

$$S_{k,n-1}^K + 2\theta^W + \psi \ge E_{k,n+1}^K - \eta \cdot \left(R_{k,n+1} + 1 - \sum_{i,n' \in \mathbf{N}^I, l \in \mathbf{L}^2} X_{i,n',k,n,l} \right) \quad \forall k, n \in \mathbf{N}^K \backslash \{1, N^{maxk}\} \tag{38}$$

Constraints (37) exclude schedules that have a long route succeeding a short route directly, and violate the working time limit, as depicted in Fig. 8a. Specifically, if slot n of driver k is the first segment of a long route (the summation term being 1) and it is started without resting ($R_{k,n} = 0$), then the end time of slot $n+1$ should be less than the start time of slot $n-1$ plus $2\theta^W + \psi$. Constraints (38) follow the same idea, for the case of a short route succeeding a long route directly.

$$E_{i,n,k,n',l,j} = S_{i,n,k,n',l,j} + \vartheta_j \cdot X_{i,n,k,n',l} + \omega_{i,j} \cdot F_{i,n,k,n',l,j} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \tag{39}$$

$$S_{i,n,k,n',l,j'} \ge E_{i,n,k,n',l,j} + \tau o_{j,j'} - \eta \cdot (1 - X_{i,n,k,n',l}) \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, (j, j') \in \mathbf{A}_l \tag{40}$$

$$S_{i,n,k,n'+1,l',j'} \ge E_{i,n,k,n',l,j} + \tau o_{j,j'} + \psi - \eta \cdot (1 - X_{i,n,k,n',l}) \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l \in \mathbf{L}^1, l' \in \mathbf{L}_l^{next}, j \in \mathbf{J}_l^{end}, j' \in \mathbf{J}_{l'}^{start} \tag{41}$$

Constraints (39) relate $S_{i,n,k,n',l,j}$ with $E_{i,n,k,n',l,j}$ for the same SC node via fixed and variable working time, while constraints (40) relate these two variables for the two consecutively visited SC nodes using the travel time parameter $\tau o_{jj'}$. Note that the variable delivering time is considered in constraints (39), where $\omega_{i,j}$ is the reciprocal of the rate of delivery. Constraints (41) state that the start time of the second segment of a long route, l', should be greater than the end time of the first segment, l, plus resting time, plus the travel time from the last SC node of l to the first SC node of l'.

$$S_l^L = \sum_{i,n \in \mathbf{N}^I, k, n' \in N^K, j \in \mathbf{J}_l^{start}} S_{i,n,k,n',l,j} \quad \forall l \tag{42}$$

$$E_l^L = \sum_{i,n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, j \in \mathbf{J}_l^{end}} E_{i,n,k,n',l,j} \quad \forall l \tag{43}$$

$$E_l^L \le S_l^L + \theta^W \quad \forall l \tag{44}$$

$$E_{l'}^L \le S_l^L + 2\theta^W + \psi \quad \forall l \in \mathbf{L}^1, l' \in \mathbf{L}_l^{next} \tag{45}$$

Constraints (42)/(43) define the segment start/end time variables S_l^L/E_l^L. Constraints (44) and (45) express restrictions on the durations of a single-route segment and a long route with two segments.

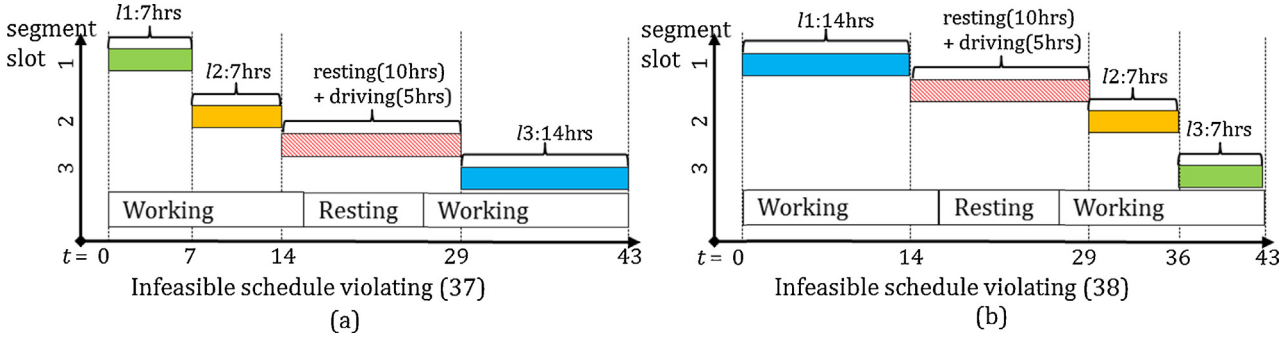

Fig. 8. Illustration of infeasible schedules that are cut off by (37) and (38). For both cases, the resting time limit is 10 h, while the maximum daily working time limit is 14 h.

5.5. Delivery flow constraints

Delivery flow should respect truck capacities, as well as customer demands, as follows,

$$F_{i,n,k,n',l,j} \le \xi_i \cdot X_{i,n,k,n',l} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \cap \mathbf{J}^C \tag{46}$$

$$F^{SX}_{i,l} + \sum_{n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, j \in \mathbf{J}_l \cap \mathbf{J}^C} F_{i,n,k,n',l,j} = \xi_i \cdot X^{IL}_{i,l} \quad \forall i, l \in \mathbf{L}^S \tag{47}$$

$$F^{SX}_{i,l} + \sum_{n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, j \in \mathbf{J}^L_l \cap \mathbf{J}^C} F_{i,n,k,n',l,j} + \sum_{n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l' \in \mathbf{L}^{next}_l, j' \in \mathbf{J}_{l'} \cap \mathbf{J}^C} F_{i,n,k,n',l',j'} = \xi_i \cdot X^{IL}_{i,l} \quad \forall i, l \in \mathbf{L}^1 \tag{48}$$

Constraints (46) enforce no product delivery when $X_{i,n,k,n',l} = 0$. Truck capacity constraints are expressed in constraints (47) and (48), respectively for short and long routes. In constraints (48), the two summations represent the delivery amount on the first and the second segments of a long route.

$$F^{LJ}_{l,j} = \sum_{i, n \in \mathbf{N}^I_i, k, n' \in \mathbf{N}^K_k} F_{i,n,k,n',l,j} \quad \forall l, j \in \mathbf{J}_l \tag{49}$$

$$\sigma^{\mathrm{MIN}}_j \le \sum_{l \in \mathbf{L}_j} F^{LJ}_{l,j} \le \sigma^{\mathrm{MAX}}_j \quad \forall j \in \mathbf{J}^C \tag{50}$$

Constraints (49) define the segment-customer aggregated delivery flow variable $F^{LJ}_{l,j}$. Constraints (50) state that the total delivery amount to a customer should satisfy its minimum and maximum demands.

5.6. Access window constraints

Each visit to a customer should be within one of the customer access windows, as follows,

$$\sum_m W_{i,n,k,n',l,j,m} = X_{i,n,k,n',l} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \cap \mathbf{J}^C \tag{51}$$

$$S_{i,n,k,n',l,j} \ge \sum_m \sigma^{AS}_{j,m} \cdot W_{i,n,k,n',l,j,m} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \cap \mathbf{J}^C \tag{52}$$

$$E_{i,n,k,n',l,j} \le \sum_m \sigma^{AE}_{j,m} \cdot W_{i,n,k,n',l,j,m} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \cap \mathbf{J}^C \tag{53}$$

Constraints (51) require that if segment l is assigned to a truck and a driver, then the visit to a customer should correspond to an access window. Constraints (52) and (53) enforce access window restrictions.

5.7. Inventory constraints

When the consumption rate is constant, constraints in this subsection are used for inventory bounds, as follows,

$$\sum_{n \in \mathbf{N}^J_j} Y_{l,j,n} = \sum_i X^{IL}_{i,l} \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A \tag{54}$$

$$\sum_{l \in \mathbf{L}_j} Y_{l,j,n} = 1 \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{55}$$

Constraints (54) state that if a segment is carried out, the visit to an anticipatable customer corresponds to one of the customer slots. Constraints (55) require that every slot of an anticipatable customer corresponds to a segment that contains this customer.

$$S^{LJ}_{l,j} = \sum_{i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K} S_{i,n,k,n',l,j} \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A \tag{56}$$

$$E^{LJ}_{l,j} = \sum_{i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K} E_{i,n,k,n',l,j} \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A \tag{57}$$

$$S^{LJ}_{l,j} - \eta \cdot (1 - Y_{l,j,n}) \le S^{JN}_{j,n} \le S^{LJ}_{l,j} + \eta \cdot (1 - Y_{l,j,n}) \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{58}$$

$$E^{LJ}_{l,j} - \eta \cdot (1 - Y_{l,j,n}) \le E^{JN}_{j,n} \le E^{LJ}_{l,j} + \eta \cdot (1 - Y_{l,j,n}) \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{59}$$

$$F^{LJ}_{l,j} - \zeta^U_j \cdot (1 - Y_{l,j,n}) \le F^{JN}_{j,n} \le F^{LJ}_{l,j} + \zeta^U_j \cdot (1 - Y_{l,j,n}) \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{60}$$

Constraints (56)/(57) define the segment-customer aggregated start/end time variables $S^{LJ}_{l,j}/E^{LJ}_{l,j}$. Constraints (58) require that if $Y_{l,j,n} = 1$, start time $S^{JN}_{j,n}$ is equal to $S^{LJ}_{l,j}$. Similar constraints are enforced for the end time and flow amount in (59) and (60).

$$L0^A_j - \rho_j \cdot S^{JN}_{j,n} + \sum_{n' < n} F^{JN}_{j,n'} \ge \zeta^L_j \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{61}$$

$$L0^A_j - \rho_j \cdot E^{JN}_{j,n} + \sum_{n' \le n} F^{JN}_{j,n'} \le \zeta^U_j \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{62}$$

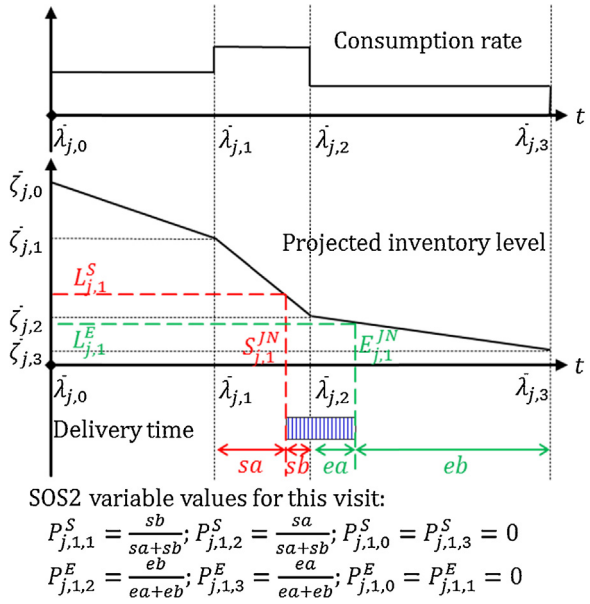

Fig. 9. Illustration of parameters and variables introduced for piecewise linear approximation, shown by an example of the first visit to customer j.

$$S^{JN}_{j,n} \geq E^{JN}_{j,n-1} \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{63}$$

Constraints (61) require that just before a delivery is made, which corresponds to one of the inventory minima during the planning horizon, the inventory should be greater than the lower bound. Constraints (62) state that inventory should be lower than the upper bound after a delivery, which corresponds to one of the inventory maxima. Constraints (63) are the sequencing constraints for visits to a customer. These constraints in conjunction with constraints (50) enforce inventory bounds throughout the horizon.

5.8. Time-varying consumption constraints

Any projected inventory level due to time-varying consumption profile can be approximated by a piecewise linear function, and modeled by special ordered set type 2 (SOS2) variables. We introduce a set of points, denoted by $q \in \mathbf{Q} = \{0, 1, \ldots, maxQ\}$, to model the projected inventory levels without deliveries. $\mathbf{Q}_j$ is the point subset for anticipatable customer j. Each $q \in \mathbf{Q}_j$ is associated with a given time $\bar{\lambda}_{j,q}$ when the consumption rate changes in the approximation, and $\bar{\lambda}_{j,0}/\bar{\lambda}_{j,maxQ}$ is the start/end time of the horizon. Each $q \in \mathbf{Q}_j$ is also associated with a projected inventory level at time $\bar{\lambda}_{j,q}$, denoted by $\bar{\zeta}_{j,q}$. Note that $\bar{\zeta}_{j,q}$ can be less than zero, because this is the inventory projection considering only consumption (no deliveries). As shown in Fig. 9, the following variables are introduced:

(a) $P^S_{j,n,q}$: SOS2 variable over index q, representing the start time of slot n of customer j; a set of SOS2 variables is defined for each (j,n) pair.
(b) $P^E_{j,n,q}$: SOS2 variable over index q, representing the end time of slot n of customer j.
(c) $L^S_{j,n}$: projected inventory level at the start of slot n of customer j (considering no deliveries).
(d) $L^E_{j,n}$: projected inventory level at the end of slot n of customer j (considering no deliveries).

The constraints are as follows,

$$S^{JN}_{j,n} = \sum_{q \in \mathbf{Q}_j} \bar{\lambda}_{j,q} \cdot P^S_{j,n,q} \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{64}$$

$$L^S_{j,n} = \sum_{q \in \mathbf{Q}_j} \bar{\zeta}_{j,q} \cdot P^S_{j,n,q} \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{65}$$

$$L^S_{j,n} + \sum_{n' < n} F^{JN}_{j,n'} \geq \zeta^L_j \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{66}$$

$$\sum_{q \in \mathbf{Q}_j} P^S_{j,n,q} = 1 \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{67}$$

In constraints (64), $P^S_{j,n,q}$ is related to $\bar{\lambda}_{j,q}$ and start time variable $S^{JN}_{j,n}$. In constraints (65), we calculate the projected inventory level at the start of slot n of customer j, based on $\bar{\zeta}_{j,q}$. Constraints (66) replace constraints (61) for the lower bound before a delivery. In constraints (67), the summation of variable $P^S_{j,n,q}$ over index q should be 1.

$$E^{JN}_{j,n} = \sum_{q \in \mathbf{Q}_j} \bar{\lambda}_{j,q} \cdot P^E_{j,n,q} \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{68}$$

$$L^E_{j,n} = \sum_{q \in \mathbf{Q}_j} \bar{\zeta}_{j,q} \cdot P^E_{j,n,q} \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{69}$$

$$L^E_{j,n} + \sum_{n' \leq n} F^{JN}_{j,n'} \leq \zeta^U_j \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{70}$$

$$\sum_{q \in \mathbf{Q}_j} P^E_{j,n,q} = 1 \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{71}$$

Constraints (68)–(71) are the counterpart of constraints (64)–(67) for the end time of a customer slot, and constraints (70) replace constraints (62) for the upper bound after a delivery.

5.9. Objective

Following the objective function (11) in the upper level VR model, we minimize the total distribution cost,

$$\min O^{SP} = \gamma^D \sum_{i,r,l \in \mathbf{L}^R_r \setminus \mathbf{L}^2} \tau^D_r \cdot X^{IL}_{i,l} + \gamma^W \sum_{i,n \in \mathbf{N}^I} (E^I_{i,n} - S^I_{i,n})$$
$$+ (\gamma^R - \gamma^W \cdot \psi) \sum_{i,l \in \mathbf{L}^1} X^{IL}_{i,l} + \gamma^V \sum_{i,l} |\mathbf{J}^L_l \cap \mathbf{J}^C| \cdot X^{IL}_{i,l} + \gamma^X \sum_{i,l \notin \mathbf{L}^2} F^{SX}_{i,l} \tag{72}$$

which includes: driving cost, working cost, resting cost, delivery cost, and penalty for unused truck capacity. The term $-\gamma^W \cdot \psi$ is included before the third summation, because the resting time during a long route is already included in the second summation.

6. Iterative approach

In the upper level VR subproblem (Section 4), we select the routes (and trucks to carry out the routes) to minimize cost; based on the selected routes, the lower level SP model (Section 5) is solved to obtain the detailed schedule. However, the selected routes can lead to infeasibility or higher distribution cost in SP, which means that multiple iterations may be needed before finding a feasible schedule and proving its optimality. Specifically, when SP is infeasible or has a higher distribution cost compared to VR, we modify the VR model by adding integer cuts and updating parameters, re-solve it to select another set of routes, and solve SP again. In this section, we present how the iterative approach is implemented.

The objective is to minimize the distribution cost, and the upper and lower bounds on this cost are provided by the solutions of the two subproblems; the penalty term for unused truck capacities is not considered. We introduce index $s \in \mathbf{S}$ to denote the iterations. The VR objective value provides a lower bound (LB) on the optimal distribution cost, since VR is a relaxed version of IRP. Thus, after solving VR, LB is updated by $LB = \max\left(LB, O^{VR} - \gamma^X \sum_{i,r} F^{RX}_{i,r}\right)$; the summation term is subtracted to exclude the penalty term for unused truck capacities. On the other hand, an upper bound (UB) on the optimal distribution cost can be obtained from the objective value of SP, since it gives a feasible solution. Similarly, UB is updated by $UB = \min\left(UB, O^{SP} - \gamma^X \sum_{i,l \notin \mathbf{L}^2} F^{SX}_{i,l}\right)$. When LB and UB are close enough or when a predefined iteration number is reached, i.e., $(UB - LB)/LB \leq \varepsilon$ or $s = s^{\mathrm{MAX}}$, the algorithm terminates. Note that both LB and UB correspond to the problem we consider after the dynamic network reduction.

The fundamental reason that the iterative approach may require multiple iterations is because the upper level problem is a relaxation of IRP; drivers are not modeled explicitly, and inventory levels are not monitored over time. Thus, we may need to iterate in the following cases:

(a) No integer feasible solution can be found by SP, because (i) there are not enough drivers to carry out the routes selected in VR (since drivers are not considered in VR); or (ii) some routes are not feasible for SP when scheduling constraints are considered.
(b) The solution of SP has a higher cost compared to VR, because for some routes selected in VR, longer working time is needed.

To address these cases, we can add integer cuts or update parameters. There are multiple options to modify VR, before re-solving it. One approach is to simply add "no-good" integer cuts (Section 6.1), which may lead to intractable iterations (Hooker et al., 2000; Harjunkoski and Grossmann, 2002; Maravelias, 2006). To reduce the number of iterations, we can also use some heuristics. More specifically, we can employ one of these three procedures, depending on the SP solution (Section 6.3 and Section 6.4):

(a) Add route number constraints if SP is integer infeasible due to the number of drivers, or
(b) Add heuristic integer cuts if SP is integer infeasible due to the routes that lead to infeasibility, or
(c) Update parameters if SP is feasible but $UB > LB$.

As shown in procedures (a) and (b), the infeasibility of SP is due to either the number of drivers or infeasible routes; this reason can be identified by solving a modified SP model with slack variables (SPS).

Another option is to generate different SP models using the current VR solution (in Section 6.2). Upper level VR decides the routes to select, as well as the truck-route pairings. The latter decision can be either enforced or relaxed when generating the lower level SP. Enforcing truck-route pairings leads to a smaller model and faster solution time for SP. On the other hand, relaxing truck-route pairings can potentially reduce the number of iterations, through more effective integer cuts (on condition that the resulting SP model can be solved fast enough). The overall solution method is summarized in Fig. 10.

6.1. General integer cuts for VR

If the iterative procedure is not terminated after solving SP, i.e., if SP is infeasible, or if UB is greater than LB, we need to add integer cuts to cut off the current VR solution. We introduce set $\mathbf{R}^G_{s,i}$ denoting the route carried out by truck i in iteration s. In other words, $\mathbf{R}^G_{s,i} = \{r | Z_{i,r} = 1\}$, where the value of $Z_{i,r}$ is from the VR solution in iteration s. Previous solutions can be avoided by adding the following "*no-good*" integer cut,

$$\sum_{i,r \in \mathbf{R}^G_{s,i}} Z_{i,r} - \sum_{i,r \notin \mathbf{R}^G_{s,i}} Z_{i,r} \leq \sum_i |R^G_{s,i}| - 1 \quad \forall s \tag{73}$$

Note that this inequality only cuts off the exact truck-route selections, which may make the iterative procedure lengthy. To reduce the number of iterations, more effective procedures to avoid symmetric solutions are proposed in the following three subsections.

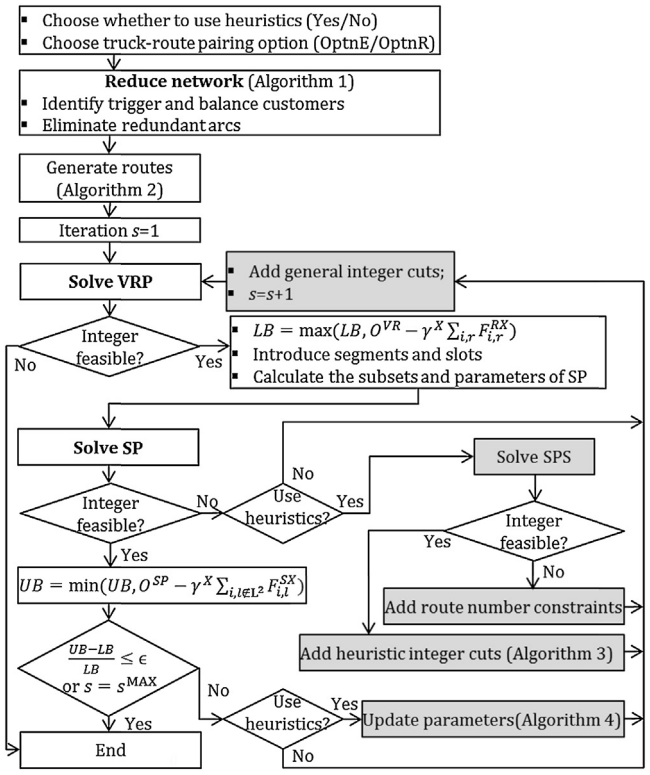

Fig. 10. Detailed solution method flowchart; diamonds represent decision points, white boxes represent the main procedures, and grey boxes represent procedures to run before re-solving the upper level VR model. Algorithms 1–4 are shown in Appendix A.

6.2. Truck-route pairing options

Binary variable $Z_{i,r}$ determines whether truck i is assigned to route r. If route r is selected by any truck, its related segments are generated for the lower level SP. As introduced earlier, $\mathbf{I}_l$ denotes the set of trucks that can carry out segment l. By defining $\mathbf{I}_l$ differently, we have the flexibility to choose if the truck-route pairings are enforced in SP. The following two options of defining subset $\mathbf{I}_l$ will be referred as *OptnE*/*OptnR*, standing for enforced/relaxed truck-route pairing option.

In OptnE, subset $\mathbf{I}_l$ is defined as follows,

$$\mathbf{I}_l = \left\{ i \,\middle|\, \sum_{r \in \mathbf{R}_l} Z_{i,r} > 0 \right\} \tag{74}$$

which means that segment l can be carried out by truck i in SP, only if the route related to l is assigned to truck i in VR.

In OptnR, we relax some of the truck-route pairings. The rule is as follows: if truck i carries out more than one route in VR, i.e., if $\sum_{r \in \mathbf{R}} Z_{i,r} > 1$, then the routes carried out by this truck can be

assigned to other trucks in SP; however, if truck i carries out exactly one route, then this route is assigned to truck i in SP, as follows,

$$\mathbf{I}_l=\begin{cases}\left\{i\left|\sum_{r\in\mathbf{R}_l}Z_{i,r}>0\right.\right\} & \text{if } \exists i\in\mathbf{I}:\sum_{r\in\mathbf{R}_l}Z_{i,r}=1 \text{ and } \sum_{r\notin\mathbf{R}_l}Z_{i,r}=0\\ \mathbf{I} & \text{otherwise}\end{cases} \quad (75)$$

Each of these two options have advantages and disadvantages. OptnE leads to a smaller SP model and faster solution time; while OptnR requires fewer iterations, because relaxing the truck-route pairing can avoid some infeasibilities. Also, stronger integer cuts may be used with OptnR, as we discuss next.

6.3. Heuristic procedures for infeasible SP

When no integer feasible solution is found by SP, there are two possible reasons: either there are not enough drivers to carry out the selected routes, or some routes are infeasible (even if there were enough drivers). By solving SP with slack variables for access window and inventory bound violations, we can identify which reason leads to infeasibility. The following non-negative variables are introduced:

(a) $\hat{S}_{i,n,k,n',l,j}/\hat{E}_{i,n,k,n',l,j}$: the violation in the start/end time to visit SC node j using truck-slot (i,n) and driver-slot (k,n') on segment l.
(b) $\hat{F}^L_{j,n}/\hat{F}^U_{j,n}$: the violation in the inventory lower/upper bound of customer j on slot n.

Using these slack variables, constraints (52), (53), (61), (62) are replaced by constraints (76)–(79),

$$S_{i,n,k,n',l,j}+\hat{S}_{i,n,k,n',l,j}\geq\sum_m\sigma^{AS}_{j,m}\cdot W_{i,n,k,n',l,j,m}$$
$$\forall i,n\in\mathbf{N}^I_i,k,n'\in\mathbf{N}^K_k,l,j\in\mathbf{J}_l\cap\mathbf{J}^C \quad (76)$$

$$E_{i,n,k,n',l,j}-\hat{E}_{i,n,k,n',l,j}\leq\sum_m\sigma^{AE}_{j,m}\cdot W_{i,n,k,n',l,j,m}$$
$$\forall i,n\in\mathbf{N}^I_i,k,n'\in\mathbf{N}^K_k,l,j\in\mathbf{J}_l\cap\mathbf{J}^C \quad (77)$$

$$L0^A_j-\rho_j\cdot S^{JN}_{j,n}+\sum_{n'<n}F^{JN}_{j,n'}+\hat{F}^L_{j,n}\geq\zeta^L_j\quad\forall j\in\mathbf{J}^A,n\in\mathbf{N}^J_j \quad (78)$$

$$L0^A_j-\rho_j\cdot E^{JN}_{j,n}+\sum_{n'\leq n}F^{JN}_{j,n'}-\hat{F}^U_{j,n}\leq\zeta^U_j\quad\forall j\in\mathbf{J}^A,n\in\mathbf{N}^J_j \quad (79)$$

The new model, which consists of constraints (20)–(51), (54)–(60), (63), (76)–(79), and minimizes objective function (72) with penalty terms for the slack variables, is referred to as model SPS.

Therefore, if SP is integer infeasible, we solve SPS. If SPS is integer infeasible, then the number of drivers is not enough to carry out the selected routes in the planning horizon; otherwise, if SPS is integer feasible, then some slack variables are greater than zero, and the corresponding routes lead to access window or inventory bound violations. For the former case, we add the route number constraints (80)–(82) below, and re-solve VR; for the latter, we identify the infeasible routes and add the corresponding heuristic integer cuts, before VR is re-solved.

If SPS is integer infeasible, these route number constraints are added:

$$\sum_{i,r:\tau^R_r\geq\eta/2}Z_{i,r}\leq|\mathbf{K}| \quad (80)$$

$$\sum_{i,r}\tau^W_rZ_{i,r}\leq|\mathbf{K}|\cdot\left\{\left\lfloor\frac{\eta}{24}\right\rfloor\theta^W+\min(\theta^W,\eta\bmod 24)\right\} \quad (81)$$

$$\sum_{i,r}\tau^D_rZ_{i,r}\leq|\mathbf{K}|\cdot\left\{\left\lfloor\frac{\eta}{24}\right\rfloor\theta^D+\min(\theta^D,\eta\bmod 24)\right\} \quad (82)$$

In constraint (80), the total number of selected routes that are longer than half of the horizon should be less than or equal to the number of drivers. In constraint (81), the summation of working time over the selected routes should be less than the summation of maximum working time over drivers; the term in the curly brackets is the maximum working time of one driver in the planning horizon. Constraint (82) is the counterpart of constraint (81) for driving time.

If SPS returns an integer feasible solutions, then there are two possible reasons:

(a) Inventory levels are violated in the detailed scheduling problem. For example, a customer initially has comparatively high inventory, and the consumption rate is quite large. Thus, it needs to be served after a certain time so that the demand and inventory upper bound can be respected at the same time. However, this customer must be visited earlier using routes selected in the VR solution.
(b) When some customers have overlapping or strict access windows, especially when they have multiple windows, it is infeasible to have them scheduled in a certain sequence, despite the preliminary filtering done by constraints (16) and criterion (c) when generating routes.

Based on the non-zero slack variables in SPS, we can identify the routes that lead to the infeasibility, and add integer cuts to the VR model. The procedure is summarized in Algorithm 3 in Appendix A, and the heuristic integer cuts are generated based on the truck-route pairing option. If OptnE is adopted, we introduce infeasible truck set $\mathbf{I}^E_s$ and infeasible route set $\mathbf{R}^E_{i,s}$ (determined in Algorithm 3), and add the following constraints

$$\sum_{r\in\mathbf{R}^E_{i,s}}Z_{i,r}\leq|\mathbf{R}^E_{i,s}|-1\quad\forall s,i\in\mathbf{I}^E_s \quad (83)$$

to exclude the infeasible route combinations for the assigned truck. Otherwise, if OptnR is used, we introduce infeasible route set $\boldsymbol{R}^R_s$, and add the following constraints

$$\sum_{i,r\in\mathbf{R}^R_s}Z_{i,r}\leq|\mathbf{R}^R_s|-1\quad\forall s \quad (84)$$

to exclude the infeasible route combinations for all trucks.

6.4. Heuristic procedures for feasible SP

If SP is feasible but $UB>LB$, it means that the cost for executing some routes in SP is higher than that in VR (which was precalculated). This is due to longer working time needed in SP, if the inventory or access window constraints require additional waiting at customers. We introduce another parameter, $\tau x_{i,r}$, representing the extra working time needed for truck i to carry out route r in SP; $\tau x_{i,r}$ is initially set to zero, and updated after solving SP in each iteration. Because a route may be assigned to more than one truck, the extra driving time for different trucks to carry out the same route can be different (even for using OptnR). Thus, this parameter update does not depend on the truck-route pairing option. After updating

Table 1
Different options in the iterative approach.

Option	1	2	3	4
Truck-route paring option	OptnE	OptnE	OptnR	OptnR
Heuristics option	No	Yes	No	Yes

$\tau x_{i,r}$ from the SP solution, objective function (11) and constraints (15) of the original VR model are modified as follows,

$$\min \sum_{i,r} \left[(\gamma_r^R + \gamma^W \cdot \tau x_{i,r}) Z_{i,r} + \gamma^X F_{i,r}^{RX} \right] \tag{85}$$

$$\sum_r (\tau_r^R + \tau x_{i,r}) Z_{i,r} \le \eta, \quad \forall i \tag{86}$$

Algorithm 4 in Appendix A summarizes the parameter updating procedure.

7. Computational study

In this section, we first use a toy example to illustrate the different options of solution methods, and then we present results based on industrial-size instances. For all instances, the horizon is 48 h, check-in/out time is 0.5 h, loading/delivering time at the plant/customers is 1 h, minimum resting time is 10 h, and maximum daily driving/working time is 11/14 h. The unused capacity penalty is \$0.1 per unit, and other cost parameters are: driving cost γ^D = \$40/hour, working cost γ^W = \$8/hour, visit cost γ^V = \$10/visit, rest cost γ^R = \$100/rest. The 48-h horizon is chosen based on industrial requirements as well as an analysis of the benefit obtained from using a horizon longer than two days. Using a shorter horizon can lead to myopic solutions, while using a longer horizon will lead, in general, to computationally hard problems with uncertain returns since the uncertainty beyond 48 h increases significantly. We tested all the problems using 4 different options (combinations of truck-route parings and heuristics), as summarized in Table 1.

All the models and solution methods were implemented in GAMS 24.7 and solved using CPLEX 12.6.3.0 on a desktop with a 3.4 GHz Intel Core processor (i7-2600) and 8GB RAM on Windows 7. The solution time limit was set to 300 s for each mathematical program. The termination criterion, ε, was 0.005. Also, the iterative procedure was terminated after 20 iterations.

7.1. Toy example

We consider an example with 3 customers, 5 trucks and 6 drivers. This example was fabricated to illustrate the complexities that may be present and that we should account for. The network structure is shown in Fig. 11, the data for customers and trucks are given in Tables 2 and 3, and iterations and solution time are summarized in Table 4.

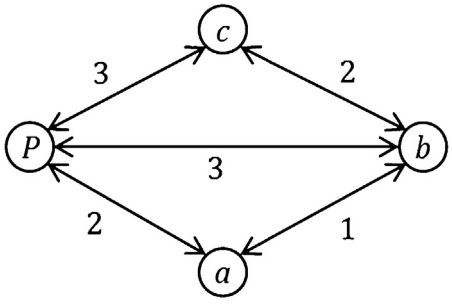

Fig. 11. Network structure for the toy example. *P* is the plant, and *a*, *b*, *c* are three customers.

Table 2
Customer parameters for the toy example.

Customer	*a*	*b*	*c*
Consumption per hour	4	6	10
Min/max level	0/400	0/500	0/850
Safety/initial level	160/200	200/300	340/350
Access window	[0,7][40,48]	[0,48]	[0,7][40,48]

Table 3
Truck capacities for the toy example.

Truck	T1	T2	T3	T4	T5
Capacity	600	1100	1100	1100	600

Table 4
Iterations and solution time for the toy example.

Option	1	2	3	4
Iterations	7	7	7	3
Time (s)	6.6	10.4	6.5	4.5

There are several optimal solutions for this problem (i.e., solutions with the same objective function value). In one of the optimal solutions, truck T1 takes route $P \to c \to P$, arrives at customer *c* at time 6, and delivers 570 units of product; truck T2 takes route $P \to b \to a \to P$, arrives at customer *b*/*a* at 42.5/44.5, and delivers 448/152 units of product. The objective function value is 665. It takes 48 min to solve this toy problem and prove optimality using a full IRP model (Dong et al., 2014), while this optimal solution is found within seconds using the proposed decomposition method, even though multiple iterations are needed.

First, we discuss the iterations using option 1. The most economic truck-route selection in the upper level VR subproblem would be that one truck with a capacity of 1100 (T2, T3 or T4) serves all three customers in a single route with no driver rest, delivering to *a*, *b*, *c* respectively 152, 478, 470 units of product; with a VR objective value O^{VR} = 454. However, routes $P \to a \to b \to c \to P$ or $P \to c \to b \to a \to P$, would lead to an infeasible SP, because the access window constraints and the inventory lower bounds cannot be satisfied at the same time. It takes 6 iterations to exclude the (symmetric) infeasible truck-route selections, that is, in iterations 1-6 trucks T2, T3, T4 take routes $P \to a \to b \to c \to P$ or $P \to c \to b \to a \to P$, and the *LB*/*UB* are 454/+∞. In the VR subproblem in iteration 7, one truck with a capacity of 600, T1 or T5, delivers to *a* and *b* 152 and 448 units respectively, and one truck with capacity of 600 visits *c*. Thus, $LB = O^{VR}$ = 662. This truck-route selection is feasible in SP, but due to customer capacity and window restrictions, only 570 out of 600 can be delivered to *c*; thus, O^{SP} is 665, and the *UB* is updated to 662 (because of the exclusion of the penalty term). Since *UB* and *LB* converge, the solution process ends at iteration 7.

Using options 2 and 3 leads to the same iterations as when using option 1. As can be seen in Table 2, more solution time is needed for option 2, because it includes the additional model SPS to solve. Finally, 2 iterations are needed to exclude the selections of routes $P \to a \to b \to c \to P$ and $P \to c \to b \to a \to P$, when option 4 is used. In iteration 3, routes $P \to c \to P$ and $P \to b \to a \to P$ are selected, and the iterative procedure ends ($LB = O^{VR}$ = 662, O^{VR} = 665, *UB* = 662).

7.2. Industrial-size instances

We consider 12 instances based on real industrial cases, with 45 to 155 customers in the original networks (including 2 to 11 order-only customers). After the dynamic network reduction (Section 3), there are typically fewer than 35 customers (including 0–2 order-only customers). The parameters used in each instance are given in the supplementary material. We classify the 12 instances into 3

groups, based on the number of selected customers: instances 1–4 have 5–14 selected customers; instances 5–8 have 15–24, while instances 9–12 have 25–34. Generally speaking, more selected customers lead to a larger problem. Four options were used for our testing. Table 5 shows the overall algorithm performance, including instance sizes, iteration numbers, total solution time and objective values. Model statistics are shown in Tables 6–8, where the VR and SP models in the first iteration are shown as representatives. Note that the statistics of the VR model in the first iteration are all the same for the four options, while the statistics of the SP model in the first iteration depend only on the truck-route paring option. We also tested instances using the full IRP model (Dong et al., 2014). The corresponding solution statistics for the smaller instances are given in Table 9.

First, we note that the decomposition method is significantly faster than the full IRP model. Using the full model, the first integer solution can only be found after a few minutes, while using the decomposition method, all instances 1–4 can be solved in a few seconds. After 20 h, the objective values of the solutions obtained by the full model are the same or inferior to the solutions obtained by the decomposition. For instances 5–12, no integer solution can be found within an hour using the full model, while all instances can be solved within 15 min using the proposed method.

We observe that SP is sometimes slow using OptnR, so OptnE should be adopted for larger problems. This is different from the toy problem, where OptnR helps to reduce the number of iterations and solution time. For large scale instances, option 2 with heuristics and OptnE is the optimal one in terms of computational cost.

For smaller problems (instances 1–4), the algorithm is finished within 10 s using all options, and the objective values are the same; option 2 is the fastest. For medium-sized problems (instances 5–8), we observe the following:

(a) OptnE is much better than OptnR, because OptnR leads to very large SP models. For example, no integer solution was found within the limit of 300 s for instance 6 using option 3. To further study this, we tested all instances with a 1200-s time limit for solving SP; OptnE still outperformed OptnR.
(b) Option 2 is the fastest; all instances were solved within 7 min. However, option 2 can lead to slightly suboptimal solutions compared to option 1, which may cut off the optimal solution in the VR subproblem (e.g., instance 6).

For larger instances, 9–12, option 2 greatly outperforms the others. Thus, using OptnE and heuristics is the best combination when obtaining near optimal solutions is acceptable (in all the instances, the gap between the solution using option 2 and the best found solution is less than 0.1%).

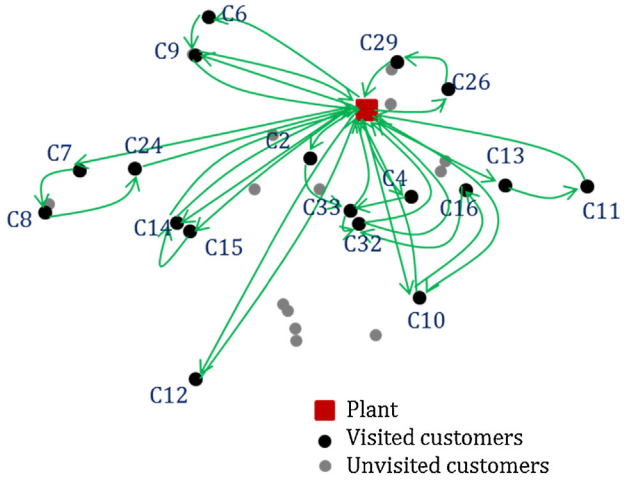

Fig. 12. Routes selected for instance 11.

Finally, we show the routing and scheduling solution of instance 11, which was also used as the example in Section 3.3. Fig. 12 shows the routing decisions (note that some balance customers are not visited in the planning horizon) and Fig. 13 shows the final solution as a Gantt chart.

7.3. Remarks

In real applications, time spent in solving IRP is critical. Thus, we discuss how to set the solution time limits for both the upper and lower level subproblems, and how to react if the time limits are reached.

For the upper level VR, we observe that the solution time depends, as expected, on problem size, but does not change greatly among iterations. For all of the tested instances, the VR model in the first iteration can be solved within 2 min, and the time increases as the numbers of trucks, arcs and routes increase (Tables 5 and 6). During the iterative procedure, integer cuts are added to VR, so the model becomes larger, but the solution time does not increase. We illustrate this observation by showing the statistics for 100 iterations of instance 11, where the VR model is solved repeatedly by adding "no-good" integer cuts. We use the results of "no-good" integer cuts, because they are the most general cuts and lead to the densest matrix. As shown in Fig. 14, even though the number of non-zeros becomes 5 times larger, the solution time does not increase significantly. Therefore, we can set a constant solution time limit for VR based on the numbers of trucks, arcs and routes. In the rare case that VR is not solved to optimality within the time limit, we should update the *LB* using the best lower bound in the

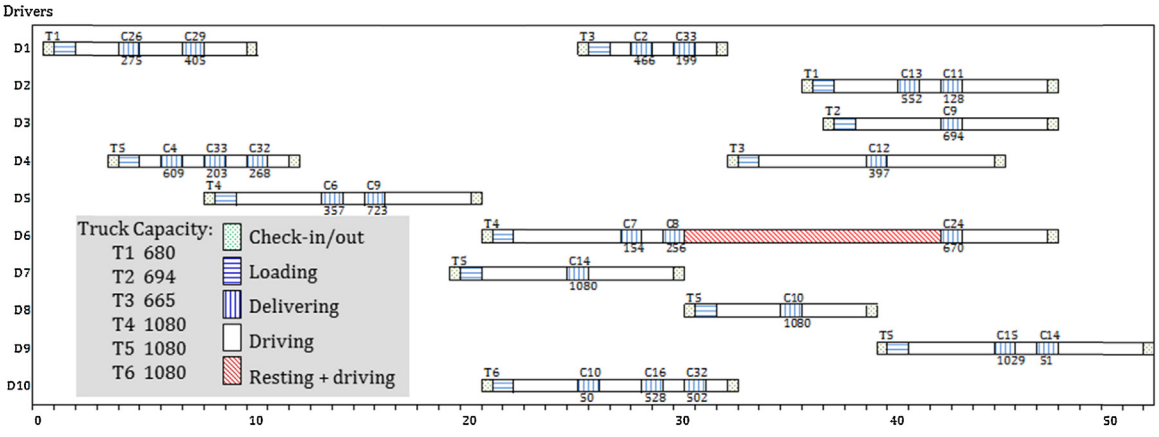

Fig. 13. Gantt chart showing the solution for instance 11.

Table 5
Instance characteristics, iterations, solution times, and objective function values using options 1–4.

Instance	Customers	Trucks	Drivers	Arcs	Routes	Iterations				Total time (s)				Objective value			
						1	2	3	4	1	2	3	4	1	2	3	4
1	5	4	4	20	17	1	1	1	1	1.4	1.3	1.4	1.3	666.0	666.0	666.0	666.0
2	7	5	5	54	49	1	1	1	1	1.9	1.8	2.1	2.1	924.0	924.0	924.0	924.0
3	8	3	5	23	16	1	1	1	1	2.6	2.6	4.0	3.9	1494.3	1494.3	1494.3	1494.3
4	13	4	6	137	218	1	1	1	1	5.4	5.6	8.4	8.4	1186.0	1186.0	1186.0	1186.0
5	16	4	6	50	40	1	1	1	1	8.1	8.1	26.0	26.5	2817.2	2817.2	2817.2	2817.2
6	17	7	9	74	100	17	2	20	2	2980.3	370.8	6105.5	970.9	5621.6	5625.6	NA	5630.9
7	23	4	6	385	1609	1	1	1	1	17.5	17.5	29.3	29.2	1809.0	1809.0	1809.0	1809.0
8	23	7	8	178	883	1	1	20	2	181.9	194.9	4840.2	881.4	5506.0	5506.0	NA	5540.5
9	25	6	9	111	112	1	1	1	1	20.9	20.9	37.4	37.3	2241.8	2241.8	2241.8	2241.8
10	32	7	10	485	2293	3	2	20	20	1372.0	878.0	6616.6	9851.9	5517.8	5517.8	NA	NA
11	32	10	13	485	4342	1	1	3	4	83.6	87.9	1773.5	3483.5	5002.8	5002.8	5002.8	5002.8
12	34	7	8	218	307	3	2	20	6	1760.2	892.1	7090.5	3952.9	3778.7	3778.7	NA	3785.8

Table 6
Solution statistics of the VR model in the first iteration.

Instance	Variables	Binaries	Constraints	Non-zeros	Nodes	Time (s)
1	248	68	226	944	1	0.08
2	650	170	593	2550	1	0.04
3	171	48	203	678	1	0.03
4	1504	360	1756	6784	1	0.08
5	512	136	636	2184	1	0.06
6	1869	448	2258	8428	1	0.11
7	16,028	3376	21,578	77,664	1	3.17
8	19,712	4130	25,993	94,373	480	7.57
9	683	171	796	2875	1	0.33
10	22,764	5054	26,491	94,584	1528	11.67
11	60,470	12,810	69,828	258,774	936	61.03
12	7224	1659	8511	29,867	1	0.39

Note: When nodes = 1, the solution was obtained and its optimality was proved, or the model was proved infeasible, in the presolve phase or at the root node.

Table 7
Solution statistics of the SP model in the first iteration, using OptnE (options 1,2).

Instance	Variables	Binaries	Constraints	Non-zeros	Nodes	Time (s)
1	243	53	324	1205	1	0.28
2	405	66	569	2060	1	0.39
3	847	207	1126	4356	1	0.67
4	2110	657	2914	10,836	1	0.39
5	3339	738	4753	17,527	1	2.70
6	18,879	3287	25,233	104,764	1275	286.11
7	3154	830	4614	16,792	1	2.82
8	6112	1406	8820	32,480	2762	153.36
9	2389	589	3303	12,740	1	3.23
10	9392	1591	13,231	50,539	932	78.25
11	8201	1540	11,285	44,042	1	22.37
12	7076	1345	9927	37,358	980	275.34

Note: When nodes = 1, the solution was obtained and its optimality was proved, or the model was proved infeasible, in the presolve phase or at the root node.

Table 8
Solution statistics of the SP model in the first iteration, using OptnR (options 3,4).

Instance	Variables	Binaries	Constraints	Non-zeros	Nodes	Time (s)
1	818	168	1043	4353	1	0.36
2	1791	254	2473	9749	1	0.38
3	1127	265	1459	6008	1	1.94
4	8571	2594	11,855	45,404	1	4.31
5	13,528	3022	18,962	71,200	1	20.76
6	60,850	9508	82,717	337,597	19	300.73
7	12,768	3514	18,226	69,062	1	14.52
8	41,608	9864	58,022	220,007	88	300.52
9	14,103	3368	19,480	78,145	1	20.12
10	94,907	11,020	94,907	368,356	1	300.15
11	14,103	3368	19,480	78,145	1	18.88
12	50,442	9727	69,732	266,266	1	300.77

Note: When nodes = 1, the solution was obtained and its optimality was proved, or the model was proved infeasible, in the presolve phase or at the root node.

Table 9
Solution statistics of the full model.

Instance	Variables	Binaries	Constraints	Non-zeros	Time of 1st integer solution (s)	Objective value of 1st integer solution	Nodes after 20 h	Objective value after 20 h	Gap after 20 h
1	16,328	10,445	8511	128,029	342	774.0	104,889	666.0	18%
2	30,852	18,260	15,576	279,301	430	1459.3	39,005	924.0	39%
3	10,130	6838	5443	81,636	52	1922.6	237,401	1496.4	11%
4	43,222	25,132	21,959	406,516	620	1734.2	22,157	1186.0	60%

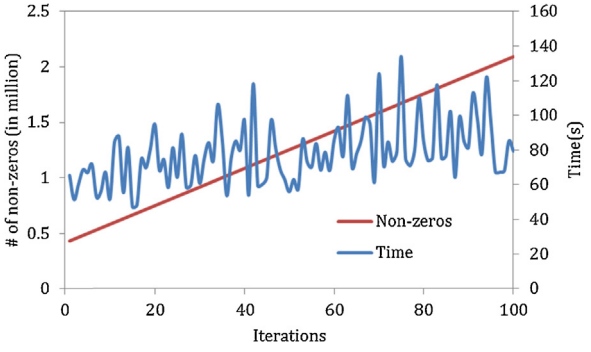

Fig. 14. Effects of integer cuts on the number of non-zeros and solution time.

branch-and-bound tree, and increase the VR time limit in the next iteration.

For the lower level SP, the size of the model depends not only on the number of trucks, customers and drivers, but also on the number of routes selected in VR in the current iteration, so the solution time can vary greatly across iterations. Accordingly, a good strategy to set the time limit for SP is to use an adaptive algorithm with the following rules: (i) The solution time limit should be a function of the numbers of trucks, customers, drivers, and selected routes in VR. (ii) When SP is not solved within the time limit, we do not use the heuristics shown in Fig. 10, and increase the time limit for the following iterations. (iii) When heuristics have been aborted and SP is solved within the time limit, we reuse the heuristics, and gradually decrease the time limit in the following iterations.

8. Conclusions

In this paper, we proposed novel solution methods for vehicle-based inventory routing problems, including a preprocessing algorithm and an iterative approach based on a decomposition to an upper level vehicle routing subproblem and a lower level detailed scheduling subproblem. The preprocessing algorithm selects *trigger* customers, whose demands should be met in the horizon, as well as *balance* customers to fully utilize truck capacities. This algorithm can be adapted to different networks by selecting user-defined parameters accordingly, and can be modified to consider different features, such as time-varying consumption rates. In the upper level subproblem, the routes to satisfy customer demand are selected, taking into account truck capacities and the working and driving time needed for each route. In the lower level subproblem, detailed truck and driver schedules are generated based on the routes determined at the upper level. We presented different types of integer cuts that can be added to the upper level problem to exclude previously found solutions or groups of solutions. Finally, we tested our methods using a set of industrial-scale instances, based on distribution networks with up to 155 customers. Instances that were intractable can now be solved within reasonable time.

Acknowledgement

The authors would like to acknowledge financial support from the US National Science Foundation under grant CBET-1264096.

Appendix A. Algorithms

Algorithm 1. Dynamic network reduction (Sections 3.1 and 3.2).

for j
  calculate parameters and subsets ζ_j^S, σ_j^{MIN}, T_j, $\mathbf{J}_j^R$
end for
$\mathbf{J}^T = \{j : \sigma_j^{\text{MIN}} > 0\}$;
for $j \in \mathbf{J}^T$
  $\mathbf{J}^B = \mathbf{J}^B \cup \left\{ j' \in \mathbf{J}_j^R \backslash \mathbf{J}^T : LO_{j'}^A - \int_0^{\eta+24T_{j'}} \rho_{j'}^T(t)\,dt < \zeta_{j'}^S \right\}$;
  if $\left\{ j' \in \mathbf{J}_j^R \backslash \mathbf{J}^T : LO_{j'}^A - \int_0^{\eta+24T_{j'}} \rho_{j'}^T(t)\,dt < \zeta_{j'}^S \right\} = \emptyset$ **then**
    $\mathbf{J}^B = \mathbf{J}^B \cup \left\{ j' \in \mathbf{J}_j^R \backslash \mathbf{J}^T : \sigma_{j'}^{\text{MAX}} = \max_{j'' \in \mathbf{J}_j^R} \sigma_{j''}^{\text{MAX}} \right\}$;
  end if
end for
$\mathbf{J} = \mathbf{J}^T \cup \mathbf{J}^B \{P\}$;
$\mathbf{A} = \mathbf{A} \backslash \left\{ (j, j') : j \notin \mathbf{J} \text{ or } j' \notin \mathbf{J} \right\}$
for $(j, j') \in \mathbf{A}$
  if (j,j') is not in the neighbor list or satisfies inequality (6) **then**
    $\mathbf{A} = \mathbf{A} \setminus \{(j, j')\}$;
  end if
  if $j \in \mathbf{J}^B$ and $j' \in \mathbf{J}^B$ and $\left\{ j'' \in \mathbf{J}^T : j \in \mathbf{J}_{j''}^R \text{ and } j' \in \mathbf{J}_{j''}^R \right\} = \emptyset$ **then**
    $\mathbf{A} = \mathbf{A} \setminus \{(j, j')\}$;
  end if
end for

Note: Lines 1–11 preprocess customers; lines 12–20 preprocess network arcs.

Algorithm 2. Routes generation for VR subproblem (Section 4.1).

declare an array *cus*[];
for *u=1:cmax*
  for *v=1:u*
    for $j \in \mathbf{J}^C$ and different from *cus*[1], *cus*[2], …, *cus*[*v*-1]
      cus[*v*] = *j*;
      if *v* = *u* **then**
        $r = P \rightarrow cus[1] \rightarrow \ldots \rightarrow cus[u] \rightarrow P$;
        calculate parameters as in equations (7)–(10);
        if *r* satisfies all the criteria in §4.1 **then**
          $\mathbf{R} = \mathbf{R} \cup \{r\}$;
        end if
      end if
    end for
  end for
end for

Note: Lines 2–7 list all possible routes; line 8 calculates the corresponding parameters; lines 9–11 verify the condition whether a route should be included in set **R**.

Algorithm 3. Set definition for heuristic integer cut generation (Section 6.3).

for $i : \sum_{n,k,n',l,j} \left(\hat{S}_{i,n,k,n',l,j} + \hat{E}_{i,n,k,n',l,j} \right) + \sum_{l,j,n} \left(\hat{F}_{j,n}^L + \hat{F}_{j,n}^U \right) X_{i,l}^{IL} Y_{l,j,n} > 0$
  for $r, l : l \in \mathbf{L}_r^R$ and $\sum_{n,k,n'} X_{i,n,k,n',l} > 0$
    if OptnE is used **then**
      $\mathbf{I}_s^E = \mathbf{I}_s^E \cup \{i\}$; $\mathbf{R}_{i,s}^E = \mathbf{R}_{i,s}^E \cup \{r\}$;
    else if OptnR is used **then**
      $\mathbf{R}_s^R = \mathbf{R}_s^R \cup \{r\}$;
    end if
  end for
end for

Note: Line 1 checks if a truck is assigned to some routes that lead to infeasibility; lines 2–10 update the sets denoting infeasible route combinations using OptE or OptR.

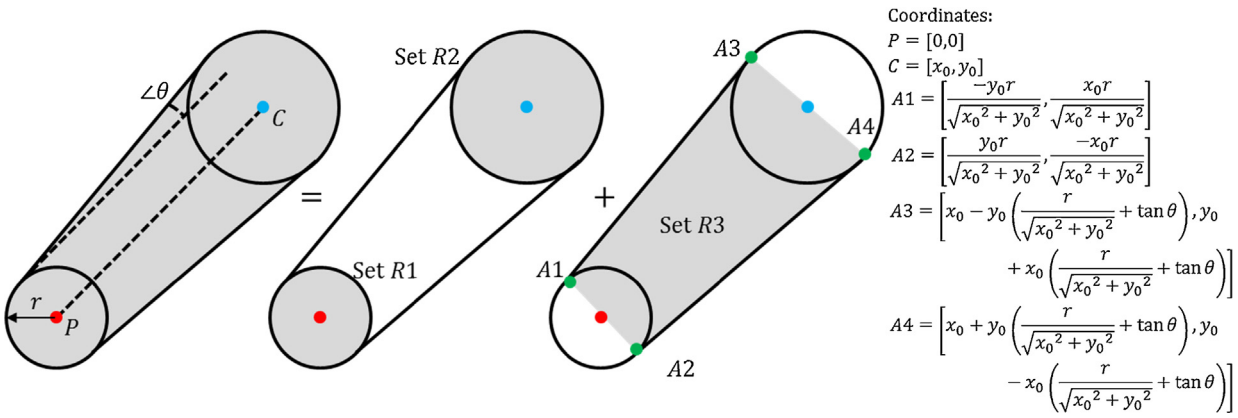

Fig. B1. The region of a trigger customer is a union of three sets.

Algorithm 4. Parameter updating for the VR subproblem (Section 6.4).

1: **for** $i, r, l : l \in \mathbf{L}_r^R \backslash \mathbf{L}^2$
2: **if** $\tau_r^R < \sum_{n,k,n'} X_{i,n,k,n',l}(E_{i,n}^I - S_{i,n}^l)$
3: $\tau x_{i,r} = \sum_{n,k,n'} X_{i,n,k,n',l}(E_{i,n}^I - S_{i,n}^l) - \tau_r^R$
4: **end if**
5: **end for**

Appendix B. Mathematical expression for trigger customer region

Let the location of the plant, P, be the origin of the Cartesian coordinate system, and $[x_0, y_0]$ denote the location of a trigger customer, C. The region of customer C (shown in Fig. 3) is a union of three sets $R1$, $R2$, $R3$. These sets are shown in Fig. B1, and defined as follows,

$$R1 = \{(x, y) | x^2 + y^2 \le r^2\}$$

$$R2 = \left\{ (x, y) \left| (x - x_0)^2 + (y - y_0)^2 \le \left(r + \tan\theta \sqrt{x_0^2 + y_0^2} \right)^2 \right. \right\}$$

$$R3 = \left\{ (x, y) \left| \begin{array}{l} 0 \le x_0 x + y_0 y \le x_0^2 + y_0^2 \\ (\tan\theta \cdot x_0 + y_0)x + (-x_0 + \tan\theta \cdot y_0)y \ge -r\sqrt{x_0^2 + y_0^2} \\ (\tan\theta \cdot x_0 - y_0)x + (x_0 + \tan\theta \cdot y_0)y \ge -r\sqrt{x_0^2 + y_0^2} \end{array} \right. \right\}$$

Appendix C. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.compchemeng.2017.02.036.

References

Adulyasak, Y., Cordeau, J.F., Jans, R., 2015. Benders decomposition for production routing under demand uncertainty. Oper. Res. 63 (4), 851–867.

Al-Khayyal, F., Hwang, S.J., 2007. Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk. Part I: Applications and model. Eur. J. Oper. Res. 176 (1), 106–130.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A., 2010. Industrial aspects and literature survey: combined inventory management and routing. Comput. Oper. Res. 37 (9), 1515–1536.

Archetti, C., Bertazzi, L., Laporte, G., Speranza, M.G., 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transport. Sci. 41 (3), 382–391.

Avella, P., Boccia, M., Wolsey, L.A., 2015. Single-item reformulations for a vendor managed inventory routing problem: computational experience with benchmark instances. Networks 65 (2), 129–138.

Aziz, N.A.B., Moin, N.H., 2007. Genetic algorithm based approach for the multi product multi period inventory routing problem. In: Proceedings of International Conference on Industrial Engineering and Engineering Management, pp. 1619–1623.

Baita, F., Ukovich, W., Pesenti, R., Favaretto, D., 1998. Dynamic routing-and-inventory problems: a review. Transport. Res. A: Pol. 32 (8), 585–598.

Bard, J.F., Nananukul, N., 2010. A branch-and-price algorithm for an integrated production and inventory routing problem. Comput. Oper. Res. 37 (12), 2202–2217.

Campbell, A.M., Savelsbergh, M.W.P., 2004. A decomposition approach for the inventory-routing problem. Transport. Sci. 38 (4), 488–502.

Coelho, L.C., Cordeau, J.F., Laporte, G., 2014. Thirty years of inventory-routing. Transport. Sci. 48 (1), 1–19.

Coelho, L.C., Laporte, G., 2015. An optimised target-level inventory replenishment policy for vendor-managed inventory systems. Int. J. Prod. Res. 53 (12), 3651–3660.

Christiansen, M., Fagerholt, K., Flatberg, T., Haugen, O., Kloster, O., Lund, E.H., 2011. Maritime inventory routing with multiple products: a case study from the cement industry. Eur. J. Oper. Res. 208 (1), 86–94.

Desaulniers, G., Rakke, J.G., Coelho, L.C., 2016. A branch-price-and-cut algorithm for the inventory routing problem. Transport. Sci. 50 (3), 1060–1076.

Disney, S.M., Potter, A.T., Gardner, B.M., 2003. The impact of vendor managed inventory on transport operations. Transport. Res. E: Log. 39 (5), 363–380.

Dong, Y., Pinto, J.M., Sundaramoorthy, A., Maravelias, C.T., 2014. MIP model for inventory routing in industrial gases supply chain. Ind. Eng. Chem. Res. 53 (44), 17214–17225.

Eppen, G.D., Martin, R.K., 1988. Determining safety stock in the presence of stochastic lead time and demand. Manage. Sci. 34 (11), 1380–1390.

Gaur, V., Fisher, M.L., 2004. A periodic inventory routing problem at a supermarket chain. Oper. Res. 52 (6), 813–822.

Goel, A., 2009. Vehicle scheduling and routing with drivers' working hours. Transport. Sci. 43 (1), 17–26.

Goel, A., 2012. The minimum duration truck driver scheduling problem. EURO J. Transp. Logist. 1 (4), 285–306.

Gounaris, C.E., Wiesemann, W., Floudas, C.A., 2013. The robust capacitated vehicle routing problem under demand uncertainty. Oper. Res. 61 (3), 677–693.

Grønhaug, R., Christiansen, M., Desaulniers, G., Descrosiers, J., 2010. A branch-and-price method for a liquefied natural gas inventory routing problem. Transport. Sci. 44 (3), 400–415.

Harjunkoski, I., Grossmann, I.E., 2002. Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. Comput. Chem. Eng. 26 (11), 1533–1552.

Hewitt, M., Nemhauser, G., Savelsbergh, M., Song, J.H., 2013. A branch-and-price guided search approach to maritime inventory routing. Comput. Oper. Res. 40 (5), 1410–1419.

Hooker, J.N., Ottosson, G., Thornsteinsson, E.S., Kim, H.-J., 2000. A scheme for unifying optimization and constraint satisfaction methods. Knowl. Eng. Rev. 15, 11–30.

Jetlund, A.S., Karimi, I.A., 2004. Improving the logistics of multi-compartment chemical tankers. Comput. Chem. Eng. 28 (8), 1267–1283.

Jiang, Y., Grossmann, I.E., 2015. Alternative mixed-integer linear programming models of a maritime inventory routing problem. Comput. Chem. Eng. 77, 147–161.

Maravelias, C.T., 2006. A decomposition framework for the scheduling of single- and multi-stage processes. Comput. Chem. Eng. 30 (3), 407–420.

Moin, N.H., Salhi, S., 2007. Inventory routing problems: a logistical overview. J. Oper. Res. Soc. 58 (9), 1185–1194.

Niakan, F., Rahimi, M., 2015. A multi-objective healthcare inventory routing problem; a fuzzy possibilistic approach. Transport. Res. E: Log. 80, 74–94.

Papageorgiou, D.J., Keha, A.B., Nemhauser, G.L., Sokol, J., 2014a. Two-stage decomposition algorithms for single product maritime inventory routing. INFORMS J. Comput. 26 (4), 825–847.
Papageorgiou, D.J., Nemhauser, G.L., Sokol, J., Cheon, M.S., Keha, A.B., 2014b. MIRPLib – a library of maritime inventory routing problem instances: survey, core model, and benchmark results. Eur. J. Oper. Res. 235 (2), 350–366.
Persson, J.A., Göthe-Lundgren, M., 2005. Shipment planning at oil refineries using column generation and valid inequalities. Eur. J. Oper. Res. 163 (3), 631–652.
Raa, B., 2015. Fleet optimization for cyclic inventory routing problems. Int. J. Prod. Econ. 160, 172–181.
Rancourt, M.E., Cordeau, J.F., Laporte, G., 2013. Long-haul vehicle routing and scheduling with working hour rules. Transport. Sci. 47 (1), 81–107.
Savelsbergh, M., Song, J.H., 2007. Inventory routing with continuous moves. Comput. Oper. Res. 34 (6), 1744–1763.
Shen, Q., Chu, F., Chen, H., 2011. A Lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. Comput. Chem. Eng. 35 (10), 2113–2123.
Singh, T., Arbogast, J.E., Neagu, N., 2015. An incremental approach using local-search heuristic for inventory routing problem in industrial gases. Comput. Chem. Eng. 80, 199–210.
Siswanto, N., Essam, D., Sarker, R., 2011. Solving the ship inventory routing and scheduling problem with undedicated compartments. Comput. Ind. Eng. 61 (2), 289–299.
Song, J.H., Furman, K.C., 2013. A maritime inventory routing problem: practical approach. Comput. Oper. Res. 40 (3), 657–665.
You, F., Pinto, J.M., Capon, E., Grossmann, I.E., Arora, N., Megan, L., 2011. Optimal distribution-inventory planning of industrial gases. I. Fast computational strategies for large-scale problems. Ind. Eng. Chem. Res. 50 (5), 2910–2927.
Yu, Y., Chu, F., Chen, H., 2006. A model and algorithm for large scale stochastic inventory routing problem. In: Proceedings of Service Systems and Service Management International Conference, pp. 355–360.
Zhang, Q., Sundaramoorthy, A., Grossmann, I.E., Pinto, J.M., 2017. Multiscale production routing in multicommodity supply chains with complex production facilities. Comput. Oper. Res. 79, 207–222.