Modeling and Solution Methods for Inventory Routing Problems in Chemical Industry

By

Yachao Dong

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Chemical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2017

Date of final oral examination: 04/24/2017

The dissertation is approved by the following members of the Final Oral Committee:

Christos T. Maravelias, Professor, Chemical and Biological Engineering

James B. Rawlings, Professor, Chemical and Biological Engineering

Ross E. Swaney, Associate Professor, Chemical and Biological Engineering

Jeffrey T. Linderoth, Professor, Industrial and Systems Engineering

Alberto Del Pia, Assistant Professor, Industrial and Systems Engineering

Abstract

Inventory routing problem (IRP) appears in various chemical industry sectors, in which the vessel/vehicle routes and the time and amount of material deliveries are decided simultaneously. Given information about the real-time inventory levels at different nodes in the supply chain (SC), as well as their consumption rate forecasts, a central decision maker solves the IRP to minimize the total distribution cost subject to the inventory bounds. Essentially, IRP is the integration of the vehicle routing problem and the inventory management problem. Solving this integrated problem can result in significant cost savings. However, IRP is non-trivial to solve, especially for large SCs with realistic constraints considered.

First, mixed-integer programming (MIP) models are developed to address inventory routing with realistic features, including both anticipatable and order-only customers, heterogeneous vehicles, time-varying consumption rates, and driver working and resting time constraints. Valid inequalities are also presented to tighten the formulation. We also show how this model can be extended to handle other important restrictions that may appear in practice.

Second, we propose a novel method, which includes a preprocessing algorithm and a decomposition method, for solving vehicle-based IRPs. The preprocessing algorithm reduces the problem size by eliminating customers and network arcs that are irrelevant for the current horizon. The decomposition method 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 (considering drivers) is acquired using a continuous-time MIP model, by adopting the routes selected from the upper level. An iterative approach is presented based on the upper and lower level subproblems, with the addition of different types of integer cuts and parameter updates.

Third, we consider a maritime IRP (MIRP) under uncertainty. For realistic MIRP, new information (e.g. newly forecasted production/consumption rate) arrives continuously and disruptive events (e.g., delays due to bad weather) are common. Accordingly, we propose a reoptimization framework that allows us to: (1) study the impact of different sources of uncertainty on the closed-loop (i.e., implemented) solution; and (2) study how different policies impact the closed-loop cost. We show that the closed-loop cost is higher than the open-loop cost (even without uncertainty), which suggests that the methods to obtain high quality closed-loop solutions have to be studied. We also show that adopting an effective mix of polices can reduce the distribution cost greatly.

Finally, we study topics related to IRP, including terminal constraints for online scheduling and modeling of changeovers. (1) We propose new types of linear terminal constraints on inventory levels for different network structures. Compared to the traditional approach, the proposed terminal constraints can lead to better closed-loop solutions in two aspects: they prevent stockout for all types of networks we studied, and lead to savings on inventory holding cost. (2) We propose a new type of formulation for sequence-dependent changeovers, which is tighter than previously proposed formulations. Furthermore, we prove that this type of formulation is facet-defining for a certain problem. Through computational study of eight types of formulations, we show that tighter formulations do not always lead to faster solution times.

Acknowledgements

First, I want to thank my advisor, Professor Christos Maravelias, who sets a great example for me to be a dedicated researcher. He has shared with me much valuable knowledge and given me guide on the research projects. With his help, I have learnt how to think critically, how to form a research idea, how to conduct research, how to write a scientific paper, how to deliver an effective and engaging presentation; I have also become a more patient and dedicated person. This past six years will be a memory I can hold fondly later on.

I also want to thank the members of my Final Oral Examination Committee, Professors James Rawlings, Ross Swaney, Jeffrey Linderoth, and Alberto Del Pia, for reviewing this thesis, attending my defense and giving me feedbacks.

I would like to thank all the great teachers at UW Madison, who gave me very enjoyable lectures and most valuable knowledge. I want to thank Professor Christos Maravelias, Ross Swaney, James Rawling, and Dr. Eric Codner, with whom I worked as a teaching assistant; the teaching experience really helped me to discover what I want to do. I want to thank the Department of Chemical and Biological Engineering and its staff for the unfailing help and support.

I am very thankful to everyone in the research group, from whom I got a lot of friendship, help and encouragement. Members senior to me, especially Dr. Sara Velez, Andres Merchan, Murat Sen, Jee-Hoon Han, Kefeng Huang, and Rex Ng, really helped me when I started to do research. My thanks also go to Dr. Julien Granger, Jose Pinto, Arul Sundaramoorthy, Norman Jerome, Ajit Gopalakrishnan, Irene Loreto, Brian Besancon, with whom I had the opportunity to work with.

The six year experience has been most valuable to me because of many friends: some we know each other for over twenty years, some only a few months. My gratitude for the happiness, friendship and beautiful memory goes to Xuanxuan, Xiaoping, Xiaomeng, Jiaxing, Yuhai, Lisa, Ji Luo, Yaqiong, Liu Han, Dandan, Eileen, Li Mao, Xiaobo, Tim, Loren, Colin.

I am very thankful for my parents who allow me to choose what I want to do and how I want to live this life. I am also very lucky for having a big family, including my grandparents, uncles, aunts, my cousins, and my nephews and nieces, who always give me a lot of encouragement, even though I fail to keep them company for the past few years.

Yachao

April, 2017

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction[1]

1.1. Inventory Routing

To improve the efficiency of their supply chains (SCs), vendors in a wide range of manufacturing sectors switch to vendor-managed inventory practices. Vendor-managed inventory (VMI) refers to an agreement between a vendor and a customer in which the latter allows the vendor to choose the timing and volume of deliveries, while the former agrees to ensure that the customer does not run out of product. In a more traditional relationship, a customer decides the orders, and large inefficiencies can occur as a result of the timing of the orders from different customers. A switch to a VMI policy has a distinct advantage because the vendor can combine deliveries to make more efficient use of the resources and can therefore reduce the distribution cost of its SC. Since distribution costs often represent 30-50% of total operating costs, the implementation of VMI policies can have a substantial impact (Disney et al., 2003). It can also be to the advantage of the customers because the vendor can pass some of the savings to the customer, and the customer no longer has to dedicate resources to the inventory management.

However, executing a VMI policy in an effective way is nontrivial, because it requires the integration of two components of SC management: inventory control and distribution routing, which have been dealt with separately in the past. In inventory control, the goal is the determination of orders in terms of time and amount for each customer, while in distribution routing, the goal is the generation of routes (and schedules) to satisfy these orders. The integration

[1] Some of this chapter is modified from Dong et al., 2014 and Dong et al., 2017.

of the two problems, which can have a significant impact on overall system performance, leads to the inventory routing problem (IRP), which is at the heart of all VMI policies (Figure 1.1).

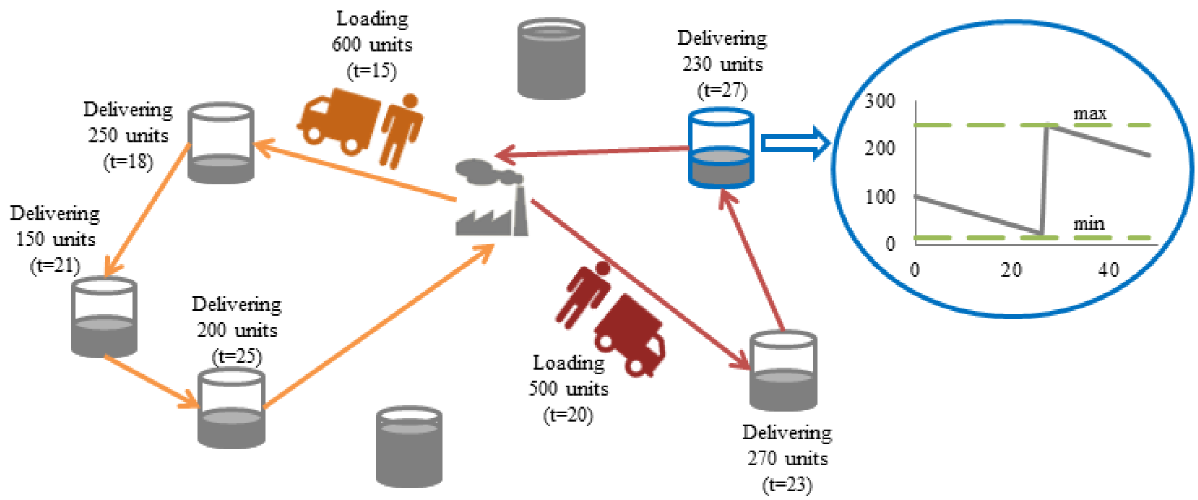


Figure 1.1. Inventory routing problem under VMI policy with drivers.

IRPs arise in a wide range of manufacturing sectors, including gas, petrochemicals, and suppliers of supermarkets and department stores (Viswanathan et al., 1997; Gaur and Fisher, 2004; Moin and Salhi, 2007; Christiansen et al., 2011; Shen et al., 2011; Singh et al., 2015). Each sector has its own special characteristics. For example, in the petroleum industry, seagoing vessels with several products in separate compartments visit multiple production and consumption ports. This special class of IRP, the so-called maritime inventory routing problem (MIRP), has been studied extensively in the literature (Jetlunt and Karimi, 2004; Savelsbergh and Song, 2007; Engineer et al., 2012).In industrial gases and retails, however, the use of trucks instead of ships introduces complex driver related constraints that are not taken into account in previous models. Furthermore, the customer consumption rate can vary significantly in a day, and some customers, such as hospitals, have strict inventory constraints. These are additional restrictions that have not been effectively modeled in previous approaches.

There are many types of IRPs studied in the literature, which can be categorized in terms of inventory policy, fleet type, and network structure (Coelho et al., 2014). When customers are visited by the vessels/trucks, 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 often shows up in MIRPs (Papageorgiou et al., 2014a; Adulyasak et al., 2015). In general, IRP can include either single-product or multi-product distribution. In the latter case, dedicated or undedicated compartments can be required (Jetlund et al., 2004; Al-Khayyal et al., 2007; Siswanto et al., 2011).

1.2. Industrial Gases Supply Chain under Vendor Managed Inventory 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 thesis, an arc means the connection between two nodes in the distribution network. A route means an ordered set of arcs, $\{a_1, a_2, \ldots, a_n\}$, in which the end node of an arc, a_i, is the same as the start node of the following arc a_{i+1}; the plant is the start node of the first arc and the end node of the last arc. 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 (Shen and Qi, 2007; You et al., 2011a; You et al., 2011b). 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 (Vidal and Goetschalckx, 1997; Verderame and Floudas, 2010). 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 thesis, we consider the short-term decisions.

1.3. Overview on Modeling Approaches

To address different types of IRPs, different mixed-integer programming (MIP) models have been proposed. The first MIP model of IRP was introduced 34 years ago (Bell et al., 1983), which includes binary variables denoting if a vehicle starts a route at a certain time and continuous variables denoting the corresponding delivery amount to a customer on the route.

For IRP under VMI, different inventory policies can be adopted. Archetti et al. (2007) modeled IRPs under three types of policies: order-up-to level policy, maximum level policy and a policy without stock upper bounds. These models include binary variables denoting if an arc is used in a certain time (i.e., if node A is followed directly after node B in the route traveled in time period t); in essence, they consider two subproblems simultaneously: the lot-sizing problem for each node, and a vehicle routing problem for each time period. However, these models assume a large fleet with homogeneous vessels, and the travelling time is not considered (i.e., it is assumed that all routes can be finished in a single period). Avella et al. (2015) considered a reformulation for similar problems with constant demand over time. In the reformulation, it is also assumed that the stock capacities are integer multiples of the demand. IRP has also been modeled in a cyclic approach (Raa, 2015), as well as using a fuzzy approach with multiple objective functions (Niakan and Rahimi, 2015). Furthermore, several consistency features have been modeled to achieve smooth operations (Coelho et al., 2012). Researchers have also proposed methods to address uncertainty (Kleywegt et al., 2002).

For maritime IRP, most of the MIP models follow a discrete-time approach. In this approach, there are binary variables denoting if a vessel starts to travel on an arc (from node A to node B) at time t, and continuous variables denoting the loading amount on a vessel when it travels on an arc (Song and Furman, 2013). Continuous-time models have also been developed by defining multiple visiting slots for each node (Jiang and Grossmann, 2015; Agra et al., 2016), where constant consumption and production rate is assumed.

1.4. Overview on Solution Methods

To solve an IRP model more effectively, different solution methods have been proposed. The solution methods are 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), , and other decomposition-based algorithms (Jetlund et al., 2004; Campbell and Savelsbergh, 2004). Heuristic based solution methods have also been developed for IRP, including genetic algorithms (Aziz and Moin, 2007), and the emulation of the logic of human planners (Ronen, 2002). The solution methods of IRP have been reviewed in several papers (Baita et al., 1998; Moin and Salhi, 2007; Andersson et al., 2010; Coelho et al., 2014; Papageorgiou et al., 2014b).

Campbell and Savelsbergh proposed a decomposition approach (Campbell and Savelsbergh, 2004). In the first phase, they determined which customers to serve on each day and how much to deliver to them; the decision was made by solving a MIP model, in which binary variables represent which route are used in each day. In the second phase, they adopted an insertion heuristic for solving vehicle routing problems with time windows, and constructed feasible vehicle routes and schedules. The time interval used for the problems in the first phase was one day, while the second phase considered a model with decision accuracy in terms of minutes.

Shen et al. proposed a Lagrangian relaxation approach for solving an IRP, in which crude oil is transported from a supply center to multiple customer harbors (Shen et al., 2011). They reformulated the capacity constraints into a non-linear form (with a maximum operator), and relaxed the constraints after introducing a set of Lagrange multipliers. They also developed a new heuristic algorithm to construct a feasible solution, based on the solution of the relaxed problem. However, their approach cannot guarantee optimality.

Song and Furman considered a MIRP which includes various practical features (Song and Furman, 2013). They proposed valid inequalities, an algorithm to implement a large neighborhood search based on a feasible integer solution, and a heuristic method based on solution polishing and the large neighborhood search.

Desaulniers et al. proposed a branch-price-and-cut algorithm for IRP (Desaulniers et al., 2016). Four types of valid inequalities were adopted, based on the minimum number of visits per customer, minimum number of routes per time interval, minimum number of subdeliveries per demand, and capacities. Column generation subproblems were solved using an ad hoc labeling algorithm.

1.5. Topics related to Inventory Routing

First, IRP is related to the traditional vehicle routing problem, in which the customers to serve and the delivery amounts are given. A lot of research effort has been made to solve different vehicle routing problems for decades (Laporte et al., 1988; Nagy and Salhi, 2007; Dondo et al., 2008; Laporte, 2009; Dondo et al., 2009; Gounaris et al., 2013). Considering the rules of drivers, models and solution methods have been developed for driver scheduling and vehicle routing problems (Goel, 2009; Goel, 2012; Rancourt et al., 2013).

Second, the integration of IRP and production has also been studied. Lei et al. were among the first researchers to consider such an integration, and they solved it through a two-phase approach (Lei et al., 2006). Glankwamdee et al. combined optimization with simulation to address the production and distribution under demand and product availability uncertainty (Glankwamdee et al., 2008). Bard and Nanaukul considered the problem to minimize the production, inventory and delivery cost simultaneously across various stages of the system, and developed a branch-and-price framework to solve the underlying MIP model (Bard and Nananukul, 2010). Zhang et al. solved a problem with multiple production facilities integrated with the inventory routing problem, and proposed a MIP model that includes production constraints in a fine time grid and vehicle routing constraints in a coarse grid; an iterative heuristic approach was developed to solve the MIP model, in which the candidate routes were updated dynamically (Zhang et al., 2017).

Third, how to write the terminal constraints is a research topic closely related to IRP. For both scheduling problems and IRPs, the optimization of a finite-horizon model to minimize the cost will push inventories to their lower bounds at the end of horizon. Therefore, such a model can only satisfy the current demand, and will overlook the consumption in future. Clearly, this situation would be suboptimal or even infeasible for the true (infinite-horizon) problem. Thus, we study how to generate terminal constraints (for scheduling problems) in this thesis.

Fourth, sequence dependent changeover in scheduling are common in the process industries (e.g., commodity, specialty, and fine chemicals; food and beverage manufacturing; pharmaceutical manufacturing; consumer goods), where cleaning-in-place, sterilization-in-place, maintenance, material transfer, and unit setup activities need to be performed between different tasks. If we view the process tasks as different nodes in the network, the changeover time is the time to travel on the arc connecting two nodes. This is very similar to inventory routing, in which the time on an arc is the travel time from one supply chain node to another. Due to the similarity, we also study the modeling of changeovers in the thesis.

1.6. Thesis Scope

The goal of the thesis is to propose a systematic framework for addressing general IRP problems. The framework includes MIP models, as well as the associated algorithms and solution techniques to find a solution in an efficient and accurate fashion. We also want to study how reoptimization should be implemented for IRP under uncertainty, as well as research topics related to IRP. More specifically, we aim to

(a) Propose MIP models that handle features and constraints that appear in IRP, especially, driver constraints.

(b) Develop solution methods that can solve IRP efficiently.

(c) Study how reoptimization should be carried out for IRP.

(d) Study related topics, including terminal constraints and changeovers.

In Chapter 2, we present MIP models to address IRP. The basic model is based on a discrete-time approach, considering both anticipatable and order-only customers, heterogeneous vehicles, and time-varying consumption rates. Valid inequalities are also proposed, as well as the constraints to account for the working, driving and resting time of drivers.

In Chapter 3, we present solution methods for vehicle-based IRPs. 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 (considering drivers) is acquired using a continuous-time MIP 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.

In Chapter 4, we study a MIRP in which shipments between production and consumption nodes are carried out by a fleet of vessels. We first propose a discrete-time MIP model. Second, we discuss different sources of uncertainty (including vessel availabilities, trip delays, pick-up window specifications, and consumption/production rate variations), and how to simulate them a rolling horizon reoptimization framework. Third, we discuss different policies that impact the quality of the closed-loop solution (including adjusting reservation windows, restricting minimum number of rented vessels, adding preference for early pick-up, and negotiating pick-up windows), and identify the optimal set of policies by using the reoptimization framework.

In Chapter 5, we propose new types of linear terminal constraints on inventory levels for online scheduling. Compared to the traditional approach, which does not exploit the relationships of inventory levels among materials, the proposed terminal constraints can lead to better closed-loop solutions in two aspects: they prevent stockout for all types of networks we studied, and lead to savings on inventory holding cost. Theoretically, we prove that for two types of networks, the proposed terminal constraints can lead to recursive feasibility.

In Chapter 6, we propose a new formulation for sequence-dependent changeovers, and prove that it is facet-defining for a certain problem. We compare the tightness of this new formulation to seven formulations that were proposed previously. Through computational study, we show that tighter formulations do not always lead to faster solution times.

Proofs and algorithms are given in the Appendices. Problem parameters and statistics can be obtained from the online supporting materials of the articles on which each chapter is based. Each chapter includes its own notation at the end.

Chapter 2

Discrete-time MIP Model for IRP under VMI Policy[2]

Different characteristics and constraints that arise in real-world have to be considered in order to make the solution of an IRP model implementable in practice. These complexities include: the coexistence of VMI customers and traditional call-in customers, the time-varying consumption profile of customers subject to their own production/sale schedules, the travelling time requirements with specified delivery time windows at customer sites, and last but not least, driver-related working time requirements, which are vital but often omitted by IRP works.

The goal of this chapter is to develop a MIP model that addresses these aforementioned challenges. Towards this goal, a discrete-time modeling approach is adopted because it can be easily modified to account for industrial restrictions. Specifically, we first propose a basic model to minimize vendor's cost while generating vehicle routes and delivery schedules. Valid inequalities to tighten the basic model are also discussed. Then, the assignment of drivers to trucks is modeled so that resting requirements for drivers are satisfied. Finally, extensions are discussed, including the modeling of inventory violations, consideration of variable loading/delivering time, additional maximum driving time constraint, and detailed modeling of drivers at the plant. While initially inspired by an industrial liquid gas transportation problem, the proposed model can be applied to other problems in the chemical manufacturing sector.

The chapter is structured as follows. In Section 2.1, we state the problem formally. In Section 2.2, a basic model is proposed based on a discrete-time approach, and then the detailed formulation and valid inequalities are presented. Section 2.3 discusses the driver-related constraints, and section 2.4 presents various extensions. Finally, section 2.5 presents an example. We use lowercase

[2] This chapter is modified from Dong et al., 2014.

italic letters for indices, uppercase bold letters for sets, and uppercase italic letters for variables. Lowercase Greek letters are used for parameters, except for parameters representing the history of the system and thus its initial state at the current planning horizon, which are denoted by upper case italic letters with a hat (e.g., $\hat{L}_{j0}^{A}$). Subsets are denoted by the letter for the superset and a superscript; e.g., $\mathbf{J}^A$ (anticipatable customers) is a subset of J (all supply chain nodes). Superscripts are also used to differentiate variables and parameters.

2.1. Problem Statement

The problem is represented by the following: a set of trucks, $i\in\mathbf{I}$; a set of SC nodes, $j\in\mathbf{J}$, which includes a central plant *P*, and a subset of customers $\mathbf{J}^C$; and a set of drivers, $k\in\mathbf{K}$. The objective is to find the optimal delivery amounts, routes, schedules, and resource allocations (drivers, trucks), to minimize the distribution cost, subject to the constraints described below. We assume that there is only one central plant, and the liquid gases are always available at the plant. It is also assumed that there is only one product in the problem, as different products use different trailers and are scheduled independently.

Each truck *i* is associated with a trailer tank of specific capacity γ_i , and the capacities can greatly vary. The customers are classified as either *anticipatable*, $j\in\mathbf{J}^A$ (i.e., customers whose inventory we can forecast and maintain), or *order-only* customers, $j\in\mathbf{J}^O$. Each customer may have multiple access windows in the horizon: given a window $m \in \mathbf{M}_j^{AH}$ during which customer *j* is accessible, we know the start/end time, $\sigma_{jm}^{AHS}/\sigma_{jm}^{AHE}$, of the window.

If traveling from *j* to *j'* is infeasible or too expensive, the arc (*j,j'*) is removed from the set of arcs $\mathbf{A} \subseteq \mathbf{J} \times \mathbf{J}$ of the SC network. The actual travel time along an arc (*j,j'*) is $\bar{\tau}_{jj'}$. The product loading time at the plant (*j*=*P*) and the delivering time at the customers ($j\in\mathbf{J}^C$), denoted by β_j, are fixed; i.e., they do not depend on the loading/delivering amount. Under this assumption, the traversal time of each arc can be calculated to include the actual travel time ($\overline{\tau}_{jj'}$) and the fixed loading/delivering time at the starting node (β_j). In section 2.4, we discuss the case in which the loading/delivering times are not fixed.

An anticipatable customer may have variable consumption rate over the planning horizon (e.g., high during the day and low or zero during the night). The usage profile is assumed to be an input (see discussion in section 2.2.1), calculated from demand forecasts prior to optimization. We are also given the capacity, ζ_j^U, of the tank and the minimum inventory level, ζ_j^L, for each anticipatable customer $j \in \mathbf{J}^A$. 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 over the current planning horizon (this assumption can easily be relaxed by introducing a set of orders, $m \in \mathbf{M}_j^O$, placed by $j \in \mathbf{J}^O$). An order from customer j is described by the amount, φ_j, and the start/end time of the window $\sigma_j^{OS}/\sigma_j^{OE}$, within which the order has to be satisfied.

For each driver, a maximum daily working time should be respected, i.e., a driver cannot work more than $\bar{\theta}^W$ hours within a 24-hour period. Also, the driver cannot work again until he has remained *off duty* for at least $\bar{\psi}$ consecutive hours. In most cases, the parameters are: $\bar{\theta}^W$= 14 hr, and $\bar{\psi}$ = 10 hr.

2.2. Basic Model

To address the majority of constraints that appear in real instances, such as variable consumption rate and multiple access hour windows, we choose to adopt a discrete-time approach. Another advantage of this approach is that it can be easily extended to account for other characteristics and constraints that the user may want to add. The disadvantages of discrete modeling of time are that it requires more variables and constraints which, given the size of this problem, leads to large MIP models and that time-related data (e.g., traveling times, loading times) have to be approximated. Nevertheless, we decided to adopt a discrete-time approach because modeling all types of problem-specific constraints is critical if an automated optimization-based tool for IRP is to be used in practice.

In this section, we show how to calculate the parameters for the discrete-time approach and we introduce a dynamic network representation of the problem. We present a MIP model that does not account for driver constraints and discuss a basic *preprocessing* algorithm for the *initialization* of the system. Valid inequalities are also presented.

2.2.1. Discrete-time Approach

The planning horizon, η, is partitioned into T time periods $t \in \mathbf{T} = \{1, 2, \ldots, T\}$ of uniform length $\delta = \eta/T$. The traversal time of arc $(j, j') \in \mathbf{A}$ is an approximation of the actual travel time and loading/delivering time, calculated by $\tau_{jj'} = \lceil (\bar{\tau}_{jj'} + \beta_j)/\delta \rceil$. We round up to ensure feasibility. For small period length, this approximation is sufficiently accurate, but in some cases the calculation may have to be modified to be less conservative. Similarly, we calculate maximum working time and minimum resting time, $\theta^W = \lfloor \bar{\theta}^W/\delta \rfloor, \psi = \lceil \bar{\psi}/\delta \rceil$. Each period t starts at point t-1, and ends at point t.

Next, we convert the original data, such as access windows, order delivery windows, and variable consumption rates, into parameters to be used in our MIP model. The calculation of the associated parameters can be performed automatically, in seconds, prior to the solution of the MIP formulation. Note that this approach leads only to changes in parameters, without increasing the size or complexity of the problem (i.e., the number of variables and constraints remain the same).

The access window time information can be obtained from the start/end time of the windows. Here, binary parameter α_{jt}^{AH} is 1 if period t is within one of the accessible windows of customer j. It is calculated by Eq (2.1), which is essentially equivalent to rounding-up the window start time and rounding-down the window end time (see Figure 2.1(a)).

$$\alpha_{jt}^{AH} = \begin{cases} 1 & \text{if } \exists m \in \mathbf{M}_j^{AH}: \lceil \sigma_{jm}^{AHS}/\delta \rceil + 1 \leq t \leq \lfloor \sigma_{jm}^{AHE}/\delta \rfloor \\ 0 & \text{otherwise} \end{cases}, \quad \forall j \in \mathbf{J}^C, t \tag{2.1}$$

Similarly for the order-only customers, binary parameter α_{jt}^{O} is 1, if period t is within the order delivery window; in Eq (2.2), it is calculated from the original window within which an order should be satisfied,

$$\alpha_{jt}^{O} = \begin{cases} 1 & \text{if } \lceil \sigma_{j}^{OS}/\delta \rceil + 1 \leq t \leq \lfloor \sigma_{j}^{OE}/\delta \rfloor \\ 0 & \text{otherwise} \end{cases}, \quad \forall j \in \mathbf{J}^O, t \tag{2.2}$$

To represent the variable consumption profile, we simply need to calculate a period-specific consumption rate ρ_{jt} for each anticipatable customer and each period, which is basically the integral of the consumption rate from t-1 to t. Figure 2.1(b) shows an example. After calculated from forecasting data prior to each optimization, the period-specific consumption rate is used in the material balance. Note that parameter ρ_{jt} can be calculated for any consumption profile function (linear, piece-wise linear, or any other non-linear).

Drivers can become available or stop working in the course of the planning horizon. The explicit availability of each driver can be calculated after the individual window bounds are rounded up or down to ensure feasibility, as presented earlier. Exactly the same approach can be followed for each truck, using the availability window bounds.

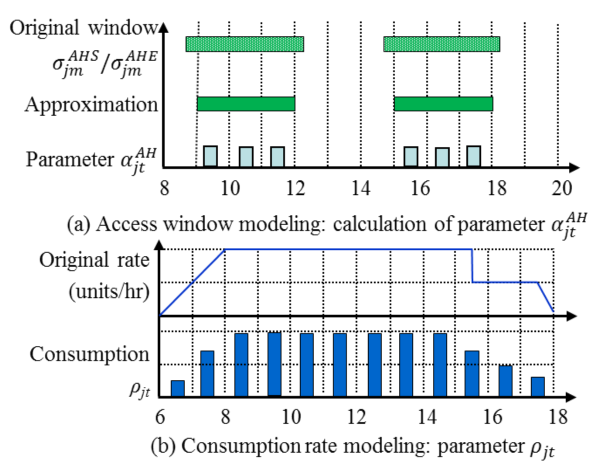


Figure 2.1. Modeling of access window and variable consumption rate (green rectangles represent access hours).

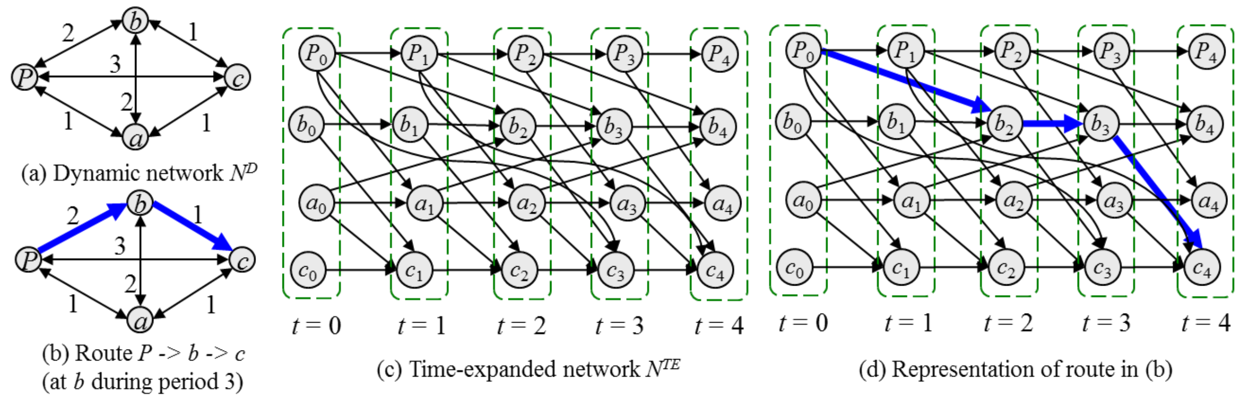


Figure 2.2. Network representation of 3-customer supply chain. (a) 3-customer supply chain represented as dynamic network. (b) An example of an incomplete truck route represented as a path in the dynamic network. (c) Time-expanded network corresponding to the one-direction (i.e., only with $P \rightarrow b$, no $b \rightarrow P$) dynamic network of figure (a). (d) Representation of the truck route in the time-expanded network.

2.2.2. Time Expanded Network Representation

The planning horizon, η, is partitioned into T time periods $t \in \mathbf{T} = \{1, 2, \ldots, T\}$ of uniform length $\delta = \eta/T$. The traversal time of arc $(j, j') \in \mathbf{A}$ is an approximation of the actual travel time and loading/delivering time, calculated by $\tau_{jj'} = \lceil(\bar{\tau}_{jj'} + \beta_j)/\delta\rceil$. We round up to ensure feasibility. For small period length, this approximation is sufficiently accurate, but in some cases the calculation may have to be modified to be less conservative. Similarly, we calculate maximum working time and minimum resting time, $\theta^W = \lfloor\bar{\theta}^W/\delta\rfloor, \psi = \lceil\bar{\psi}/\delta\rceil$. Each period t starts at point t-1, and ends at point t.

The basic model includes the following binary variables:

(a) Truck location: $\bar{X}_{ijt}$ is one if truck *i* is at SC node *j* during time period *t*.

(b) Trip start: $W_{ijj't}$ is one if truck *i* starts trip from *j* to *j'* at time point *t*.

Throughout this chapter, variables whose time index t is used to denote a state or quantity *during* a period have an over bar (e.g., $\bar{X}_{ijt}$); while variables whose index t is used to denote an action/state *at* time point *t*, have no over bar (e.g., $W_{ijj't}$). We also use the following non-negative continuous variables:

(a) Anticipatable customer inventory: L_{jt}^{A} is the inventory level of anticipatable customer j at time point t. Note that this variable stands for the inventory level of customer and truck load amount, if the truck is at the customer (see Eq (2.6), (2.7)).

(b) Arc flow: $F_{ijj't}^{A}$ is the amount of product loaded in truck i, which starts the trip from node j to node j' at time t.

Interestingly, the problem can be represented in terms of a dynamic network: the plant and the customers are the nodes, the connections between them are the arcs, and the traversal time is equal to $\tau_{jj'}$. A trip is then represented as a path in the dynamic network. A dynamic network N^{D} can be transformed into a time-expanded network N^{TE}: (1) for each node v of N^{D}, introduce T+1 nodes, v_0, v_1, …, v_T in N^{TE}, where v_t represents node v at time t; (2) introduce arc $v_t \rightarrow w_{t'}$ in N^{TE} if $v \rightarrow w$ is an arc in N^{D} and $t' - t = \tau_{vw}$. A flow along $v_t \rightarrow w_{t'}$ in N^{TE} corresponds to a flow along $v \rightarrow w$ in N^{D} (Ahiujia et al., 1993; Maravelias, 2012a). The construction of N^{TE} and its representation of a route are illustrated in Figure 2.2. Each truck has its own specific network.

2.2.3. Mathematical Formulation

The planning horizon, η, is partitioned into T time periods $t \in \mathbf{T} = \{1, 2, ..., T\}$ of uniform length $\delta = \eta / T$. The traversal time of arc $(j, j') \in \mathbf{A}$ is an approximation of the actual travel time and loading/delivering time, calculated by $\tau_{jj'} = \lceil (\bar{\tau}_{jj'} + \beta_j)/\delta \rceil$. We round up to ensure feasibility. For small period length, this approximation is sufficiently accurate, but in some cases the calculation may have to be modified to be less conservative. Similarly, we calculate maximum working time and minimum resting time, $\theta^{W} = \lfloor \bar{\theta}^{W}/\delta \rfloor, \psi = \lceil \bar{\psi}/\delta \rceil$. Each period t starts at point t-1, and ends at point t.

*Truck location***.** Variables $\bar{X}_{ijt}$ which determine the location of the truck are defined by:

$$\bar{X}_{ijt} = \bar{X}_{ij,t-1} + \sum_{j'} W_{ij'j,t-\tau_{j'j}-1} - \sum_{j'} W_{ijj',t-1}, \quad \forall i, j, t \tag{2.3}$$

where the first sum represents the arrival of truck i at customer j at time t-1, and the second sum represents the departure of truck i from j at t-1. Constraints (2.3) require that truck i is at SC node j during period t, if: (1) it was there during period t-1, and it did not leave at t-1; or (2) it arrived at node j at t-1, from a trip that started earlier, and it did not leave node j immediately. Eq (2.3) implies that a truck can be at only one node at a certain time, as long as the initial state is feasible (i.e., at t = 0 the truck is either at a node or traveling along a single arc). Note that variables $W_{ijj't}$ represent binary flows in the time expanded network for trucks and Eq (2.3) expresses a flow balance for node j_t in this network (see Figure 2.3).

*Arc flow***.** Variables $F^A_{ijj't}$, representing the truck loads along the arcs, are bounded by:

$$F^A_{ijj't} \leq \gamma_i W_{ijj't}, \quad \forall i, j, j', t \tag{2.4}$$

$$F^A_{iPjt} \geq \varepsilon \gamma_i W_{iPjt}, \quad \forall i, j, t \tag{2.5}$$

Constraints (2.4) bound the arc flow variable $F^A_{ijj't}$ by the truck capacity, if the corresponding trip start variable $W_{ijj't}$ is 1; otherwise, the arc flow variable is zero. Constraints (2.5) enforce a minimum loading amount at the plant to avoid a route with very small deliveries, where ε is a parameter less than 1. When full truck loading is required for each route, ε will be set to 1.

Anticipatable customer inventory. The inventory level L^A_{jt} is defined by:

$$L^A_{jt} = L^A_{j(t-1)} + \sum_{i,j'} F^A_{ij'j,t-\tau_{j'j}} - \sum_{i,j'} F^A_{ijj't} - \rho_{jt}, \quad \forall j \in \mathbf{J}^A, t \tag{2.6}$$

where the first sum represents the incoming flow to customer j at time point t, and the second sum represents the outgoing flow. Thus, the inventory at point t is the inventory at the previous time point, plus the incoming flow, minus the outgoing flow, minus the consumption in the last period.

The inventory level is bounded by:

$$\zeta_j^L \leq L_{jt}^A \leq \zeta_j^U + \sum_i \gamma_i \bar{X}_{ij,t+1}, \quad \forall j \in \mathbf{J}^A, t \tag{2.7}$$

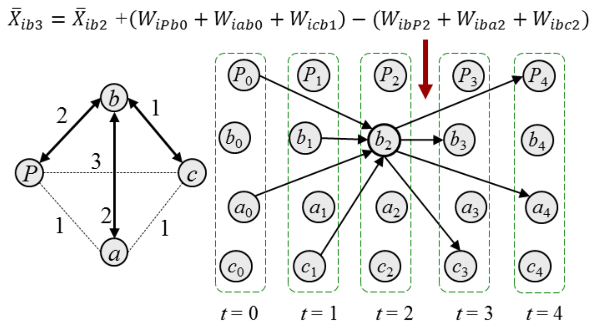


Figure 2.3. Truck location modeling, for truck *i* at customer *b* during time period 3.

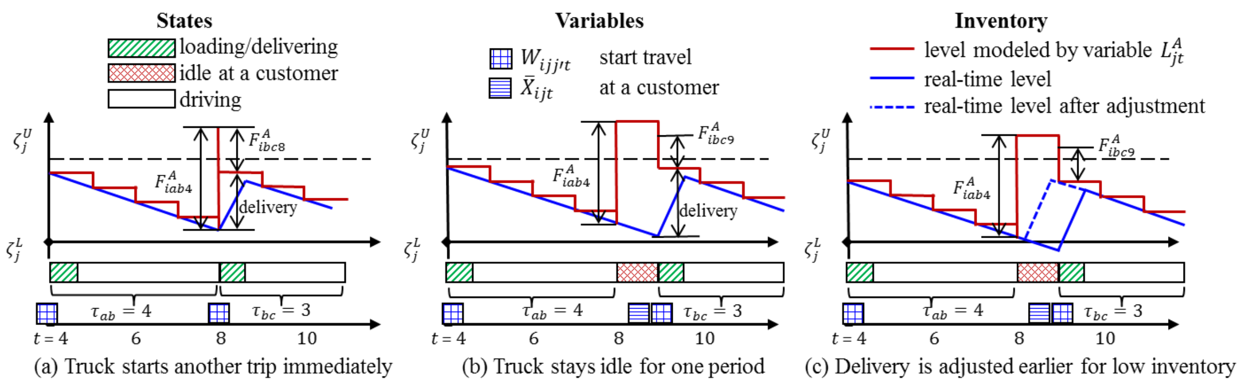


Figure 2.4. Inventory modeling for anticipatable customer *b*, with a truck traveling on *a*→*b*→*c*. The loading/delivering time is included at the beginning of each traversal time. (a) The truck arrives at *b*, immediately makes delivery, and leaves. (b) The truck stays at b, idle for one period before making delivery. (c) The real-time inventory may go below the minimum level, when the truck is at the customer but has not made delivery yet. This inventory violation can be avoided by starting the delivery earlier during the idle period, as shown in the dashed line in (c).

The lower bound is simply the minimum level, while the upper bound is the maximum level plus a summation of truck capacity multiplied by the truck location variable. Since the inventory level, L_{jt}^A, in Eq (2.6) includes the previous incoming flow for the entire tank, when truck *i* is at a customer *j*, the maximum level should be adjusted accordingly in Eq (2.7). Figure 2.4 illustrates the way we model deliveries and inventories.

Order satisfaction. For order-only customers, the order satisfaction is described by:

$$\sum_{i,j',t} F^A_{ij'jt} - \sum_{i,j',t} F^A_{ijj't} \geq \varphi_j, \quad \forall j \in \mathbf{J}^O \tag{2.8}$$

where the difference of the incoming and outgoing flows is greater than the order amount.

Objective function. The objective function minimizes cost,

$$z = min \sum_{i,j,j',t} \omega_{ijj'} W_{ijj't} \tag{2.9}$$

where $\omega_{ijj'}$ is the travel cost along the (*j,j'*) arc for truck *i*. When the cost is assumed to be proportional to the distance and thus fuel expenses, parameter $\omega_{ijj'}$ can be replaced by the actual travel time $\overline{\tau}_{jj'}$. The setup costs for each delivery can be easily modeled, by adding a sum of trip start variables $W_{ijj't}$ to the objective function. Parameters $\omega_{ijj'}$ can also capture driving time, and cost, for drivers (since the time between nodes *j* and *j'* is known) and setup cost for the head node.

2.2.4. Preprocessing

The proposed model is meant to be used in a rolling horizon fashion, which means that previously made decisions affect the current state (t = 0) of the system. Here, we assume that the initial state of the system is described by the following: (1) inventory of anticipatable customers $j \in \mathbf{J}^A$, $\hat{L}^A_{j0}$; (2) inventory of trucks, $\hat{L}^T_{i0}$; (3) truck locations ($\hat{X}_{ij0} = 1$ if truck *i* is located at *j* at t = 0, $\hat{X}_{ij0} = 0$ otherwise); (4) unfinished trips from the previous horizon ($\widehat{W}_{ijj't} = 1$, if truck *i* started trip from *j* to *j'* at *t*, $\widehat{W}_{ijj't} = 0$ otherwise); and (5) the arc flow of unfinished trips ($\hat{F}^A_{ijj't}$, = inventory of truck *i* starting a trip from *j* to *j'* at *t*). The index *t* in both $\widehat{W}_{ijj't}$ and $\hat{F}^A_{ijj't}$ parameters satisfies $-\tau_{jj'} \leq t < 0$.

Using the parameters defined above, the initial state of the system is determined as follows. For the initial location, we set $\bar{X}_{ij0} = \hat{X}_{ij0}$, for all *i, j*. For a truck on the road at *t* = 0, parameter $\widehat{W}_{ijj't}$ is included in Eq (2.3), by replacing $W_{ij'j,t-\tau_{j'j}-1}$ with $\widehat{W}_{ij'j,t-\tau_{j'j}-1}$, when $t - \tau_{j'j} - 1 < 0$. The same is done with arc flow parameter of unfinished trip $\hat{F}^{A}_{ijj't}$ in Eq (2.6). Initial inventories are set for all anticipatable customers via $L^{A}_{j0} = \hat{L}^{A}_{j0} + \sum_i \hat{L}^{T}_{i0}\hat{X}_{ij0}$, where the sum represents the truck inventory, if it is at the customer initially.

In addition, some binary variables need to be fixed. For arcs $(j, j') \notin \mathcal{A}$, the corresponding trip start variables $W_{ijj't}$ are set to zero, for all *i, t*. If time period t is not within the access window of customer *j*, i.e., $\alpha^{AH}_{jt} = 0$, $\bar{X}_{ijt}$ and $W_{ijj't}$ are set to zero, for all *i, j'*. In this way, a truck cannot stay at a customer outside the customer's access hours and the customer has no outgoing arcs outside its access window, and thereby from Eq (2.3), no incoming arcs are allowed. However, this can be relaxed, if we allow trucks to stay in a customer outside its access window. Variables $\bar{X}_{ijt}$ and $W_{ijj't}$ are dealt similarly for the order delivery windows ($\alpha^{O}_{jt} = 0$). In cases where the access window are soft, we can also model early/late deliveries with penalties. We expand the start/end time but include a penalty term for deliveries outside the window. When truck *i* is not available during time *t*, we fix $\bar{X}_{iPt}$ to one, and other $\bar{X}_{ijt}$ to zero. When truck *i* cannot serve customer *j* (e.g., due to pump type or trailer size restrictions), we can simply set $\bar{X}_{ijt}, W_{ij'jt}, W_{ijj't}$ to zero for all *j'* and *t*. Finally, if customer *j* has to be served first in a route (e.g., hospitals), then we can remove all incoming arcs except from the plant, which is equivalent to setting $W_{ij'jt}$ to zero, for all $i, j' \neq P, t$.

2.2.5. Valid Inequalities

The valid inequalities presented below, which tighten the formulation, are based on the idea of cumulative demand and maximum delivery or loading amount per visit.

For anticipatable customers, the following inequality is valid:

$$\sum_{i,j',t} W_{ij'jt} \geq \left\lceil \frac{\zeta^{L}_{j} + \sum_t \rho_{jt} - \hat{L}^{A}_{j0} - \sum_{i,j',-\tau_{j'j} \leq t < 0} \hat{F}_{ij'jt}}{\min\left(\zeta^{U}_{j} - \zeta^{L}_{j},\ \max_i \gamma_i\right)} \right\rceil, \quad \forall j \in \mathbf{J}^{A} \tag{2.10}$$

which enforces that the number of visits to a customer $j \in \mathbf{J}^A$ should be above a round-up of a lower bound, defined on the right-hand side (RHS). The numerator of the lower bound is the minimum demand in the planning horizon for this customer (minimum level at the end of horizon, plus consumption, minus initial inventory, and pre-assigned incoming arc flows), while the denominator is the maximum delivery amount per visit. Additionally, to address the impact of the finite horizon optimization approach, a terminal minimum level parameter ζ_j^T can be used, replacing ζ_j^L in the numerator. In this way, the minimum level at the end of horizon is guaranteed to be above the terminal minimum level.

Similarly, we can write the inequality for each order-only customer, with a small modification, described in the following equation:

$$\sum_{i,j',t} W_{ij'jt} \geq \left\lceil \frac{\varphi_j}{\max_i \gamma_i} \right\rceil, \quad \forall j \in \mathbf{J}^O \tag{2.11}$$

where on the RHS, the numerator is modified to the order amount, while the denominator is the maximum truck capacity.

Such inequalities can also be written for the plant as follows,

$$\sum_{i,j,t} W_{iPjt} \geq \left\lceil \frac{\sum_{j \in \mathbf{J}^A} \left(\zeta_j^L + \sum_t \rho_{jt} - \hat{L}_{j0}^A - \sum_{i,j',-\tau_{j'j} \leq t < 0} \hat{F}_{ij'jt} \right) + \sum_{j \in \mathbf{J}^O} \varphi_j}{\max_i \gamma_i} \right\rceil \tag{2.12}$$

where the numerator now represents the minimum total demand over all customers in the planning horizon.

With these valid inequalities, the basic model, **M1**, consists of Eq (2.3) –(2.12).

Note that valid inequalities in (2.10)–(2.12) have two other versions that can be useful in some cases. We will just demonstrate them for Eq (2.10), but Eq (2.11) and (2.12) can be rewritten following the same logic.

First, if the truck capacities greatly vary, we can write the valid inequality as follows:

$$\sum_{i,j',t} \min\left(\zeta_j^U - \zeta_j^L, \gamma_i\right) W_{ij'jt} \geq \zeta_j^L + \sum_t \rho_{jt} - \hat{L}_{j0}^A - \sum_{i,j',-\tau_{j'j} \leq t < 0} \hat{F}_{ij'jt}, \quad \forall j \in \mathbf{J}^A \tag{2.13}$$

where on the left-hand side (LHS) is the sum of maximum delivery amount multiplied by the incoming arc binary, with different truck capacities considered. Eq (2.13) is expected to be more effective in instances with very different truck capacities.

Second, Eq (2.10) is rewritten with respect to the minimum demand up until every time point t as follows,

$$\sum_{i,j',t' \leq t-\tau_{j'j}} W_{ij'jt'} \geq \left\lceil \frac{\zeta_j^L + \sum_{t' \leq t} \rho_{jt'} - \hat{L}_{j0}^A - \sum_{i,j',-\tau_{jj'} \leq t' \leq \min(-1, t-\tau_{j'j})} \hat{F}_{ij'jt'}}{\min\left(\zeta_j^U - \zeta_j^L,\ \max_i \gamma_i\right)} \right\rceil, \quad \forall j \in \mathbf{J}^A, t \tag{2.14}$$

where the LHS represents the number of visits to customer *j*, and the summation of consumption variables until the current time t is in the numerator on the RHS. Our computational studies show that Eq (2.14) does not typically make the formulation tighter than Eq (2.10). Similar inequalities to Eq (2.14) have been successfully implemented to address the maritime IRP.

2.3. Driver Constraints

2.3.1. New Variables

A driver is assigned to a truck, between the start time of checking-in and the finish time of checking-out. During this assignment, he could be either working or resting away from the plant. We assume that a driver that has returned to the plant can rest at his *base*. To model the driver assignments to trucks and account for working time limits, we need to differentiate between the *states* a driver can be at, so the following binary variables are introduced:

(a) Check-in: Y_{ikt}^S is one if driver *k* starts to check-in with truck *i* at time point *t*.

(b) Check-out: Y_{ikt}^F is one if driver *k* finishes the check-out process with truck *i* at time point *t*.

(c) Driver assignment: $\bar{Y}_{ikt}$ is one if driver k is assigned to truck i during time period t.

(d) Working: $\bar{Y}^W_{ikt}$ is one if driver k, assigned to truck i, is working during time period t.

(e) Resting away: $\bar{Y}^R_{ikt}$ is one if driver k, assigned to truck i, is resting away from the plant during period t.

(f) Resting at base: $\bar{V}^R_{kt}$ is one if driver k is resting at his base beyond the minimum ψ periods, during period t.

(g) Resting away (truck): W^R_{ijt} is one if the driver assigned to truck i starts a rest period after visiting customer j, at time point t.

Briefly, the first two variables represent the check-in/out activities of the driver at a certain time point, whereas the next four variables represent the states of drivers during a time period. The last one is introduced to model the truck's state when its assigned driver is resting at a certain customer site.

2.3.2. Mathematical Formulation

Truck location. As we will see below, $\bar{X}_{ijt}$ is used to monitor the on-the-clock time of a driver. This means that if a driver is resting at customer j during period t, variable $\bar{X}_{ijt}$ should be zero, as illustrated in Figure 2.5. Thus, when driver engagement is considered, Eq (2.3) should be modified for each customer site (the plant remains the same), as follows:

$$\bar{X}_{ijt} = \bar{X}_{ij,t-1} + \sum_{j'} W_{ij'j,t-\tau_{j'j}-1} - \sum_{j'} W_{ijj',t-1} - W^R_{ij,t-1} + W^R_{ij,t-\psi-1}, \quad \forall i, j \in \mathbf{J}^C, t \tag{2.15}$$

where the assigned driver can rest for ψ periods at the customer site. Since resting away from the plant leads to additional cost, the objective should be modified by adding a penalty term $\sum_{i,j,t} \omega^R_{ij} W^R_{ijt}$, where ω^R_{ij} is the driver *resting* cost for truck i at customer j.

Also, the upper bound for the inventory level variable L^{A}_{jt} needs to be modified to include the truck capacity, when the driver assigned to the truck is resting at this customer. Thus, Eq (2.7) needs to be rewritten as follows,

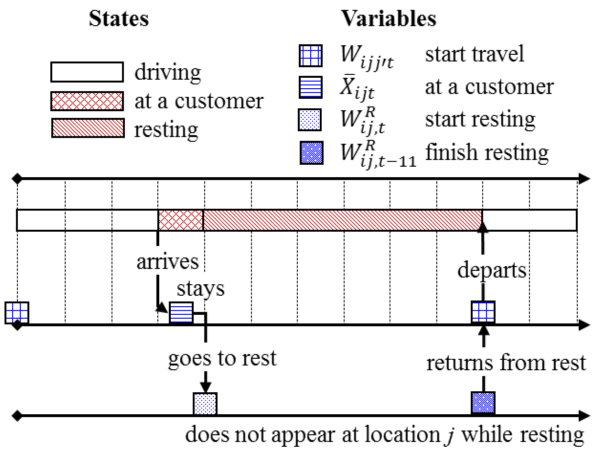


Figure 2.5. Modeling of truck location and driver rest at a customer site.

$$\zeta^{L}_{j} \leq L^{A}_{jt} \leq \zeta^{U}_{j} + \sum_{i} \gamma_{i}(\bar{X}_{ij,t+1} + \sum_{t'=t-\psi+1}^{t} W^{R}_{ijt'}), \quad \forall j \in \mathbf{J}^{A}, t \tag{2.16}$$

Driver-truck engagement. Three set of equations are introduced, to model the driver-truck engagements as follows,

$$\bar{Y}_{ikt} = \bar{Y}_{ik,t-1} + Y^{S}_{ik,t-1} - Y^{F}_{ik,t-1}, \quad \forall i, k, t \tag{2.17}$$

$$\bar{Y}_{ikt} = \bar{Y}^{W}_{ikt} + \bar{Y}^{R}_{ikt}, \quad \forall i, k, t \tag{2.18}$$

$$\sum_{k} \bar{Y}_{ikt} \leq 1, \quad \forall i, t \tag{2.19}$$

Eq (2.17) enforces that a driver is engaged with truck *i* during period *t*, if (1) he was engaged in the last period and did not check out at *t*-1, or (2) checked in at *t*-1 and did not check out immediately. Eq (2.18) enforces that when a driver is engaged with a truck, he can be either working or resting. Then, Eq (2.19) requires that each truck cannot be assigned to more than one driver.

Working and resting away. The next step is to monitor the activity of a driver through the activity of the truck the driver is assigned to. The sum of variables Y_{ikt}^{W} and Y_{ikt}^{R} are calculated as follows:

$$\sum_{k} \bar{Y}_{ikt}^{W} \geq \sum_{j,j'} \sum_{t'=t-\tau_{jj'}}^{t-1} W_{ijj't'} + \sum_{j \in \mathbf{J}^{\mathbf{A}}} \bar{X}_{ijt} + \sum_{k} \sum_{t'=t-\varphi^{CI}}^{t-1} Y_{ikt'}^{S} + \sum_{t'=t}^{t+\varphi^{CO}-1} Y_{ikt'}^{F}, \quad \forall i, t \tag{2.20}$$

$$\sum_{k} \bar{Y}_{ikt}^{R} = \sum_{j} \sum_{t'=t-\psi}^{t-1} W_{ijt'}^{R}, \quad \forall i, t \tag{2.21}$$

where $\varphi^{CI}/\varphi^{CO}$ are the check-in/check-out times respectively. Eq (2.20) enforces that if truck i is utilized during period t, then a driver assigned to it has to be working. The four terms in the RHS represent possible activities: (1) driving, (2) waiting at a customer site, (3) checking in, and (4) checking out (see Figure 2.5). Eq (2.20) is written as inequality rather than equality to include the case where the driver is assigned to a truck but is idle at the plant. Eq (2.21) requires that if someone assigned to truck i is resting away at a customer site at t, one of the associated driver resting variable $\bar{Y}_{ikt}^{R}$ should be 1.

Resting at the plant. The variable $\bar{V}_{kt}^{R}$, which represents driver resting at his base beyond the minimum ψ periods, is defined by,

$$\bar{V}_{kt}^{R} = \bar{V}_{k,t-1}^{R} + \sum_{i} Y_{ik,t-\psi-1}^{F} - \sum_{i} Y_{ik,t-1}^{S}, \quad \forall k, t \tag{2.22}$$

Eq (2.22) requires that driver k is resting (beyond the minimum time) during period t, if he has been resting or just finished the minimum resting hours after check-out, and did not check-in at the start of the current period. Eq (2.22) allows for a flexible resting time at the plant. Note that the availability of a driver is also modeled through $\bar{V}_{kt}^{R}$: if driver k is always available, the variable of the initial time period will be set to one; otherwise, if he is only available after time point t', $\bar{V}_{kt}^{R}$, Y_{ikt}^{S}, Y_{ikt}^{F}

will be set to zero for any t before t', $\bar{V}^R_{kt'}$ will be set to one, and Eq (2.22) will only be written for t greater than t'.

Maximum working time. Finally, the working time constraint is written as follows,

$$\sum_i \sum_{t' \in \mathbf{T}^{24}_t} \bar{Y}^W_{ikt'} \leq \theta^W, \quad \forall k, t \tag{2.23}$$

where $\mathbf{T}^{24}_t$ is the set of periods fully or partially included in the 24-hour interval ending with period t, which is defined by $\mathbf{T}^{24}_t = \{t' : t - 24/\delta < t' \leq t\}$.

The model with driver constraints, named **M2**, includes Eq (2.3) for $j = P$, (2.4)-(2.6), (2.8)-(2.12), and (2.15)-(2.23). Note that when Y^S_{ikt}, Y^F_{ikt} and W^R_{ijt} are required to be binary, variables $\bar{Y}_{ikt}$, $\bar{Y}^R_{ikt}$, $\bar{Y}^W_{ikt}$, and $\bar{V}^R_{kt}$ can be treated as continuous variables, bounded between 0 and 1, because constraints (2.16)-(2.21) ensure their integrality. We treat them as nonnegative continuous variables in the example shown in section 2.5.

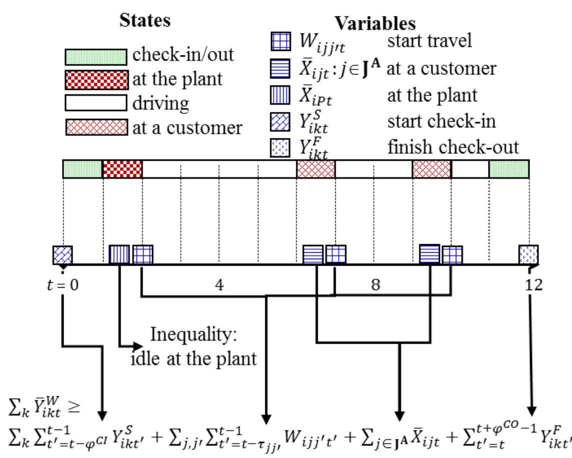


Figure 2.6. Driver working modeling, with variables corresponding to different working activities.

2.4. Extensions

In this section, we show how the proposed model can be extended to account for features that have to be considered in practice.

2.4.1. Inventory Violations

In the basic model, the inventory level of an anticipatable customer is required to be between a minimum inventory level ζ_j^L and the tank capacity ζ_j^U. However, customers often prefer to maintain their inventory above a safety level $\zeta_j^S > \zeta_j^L$. If that is the case, the following constraint is added.

$$L_{jt}^A = L_{jt}^{A+} - L_{jt}^{A-}, \quad \forall j \in \mathbf{J}^A, t \tag{2.24}$$

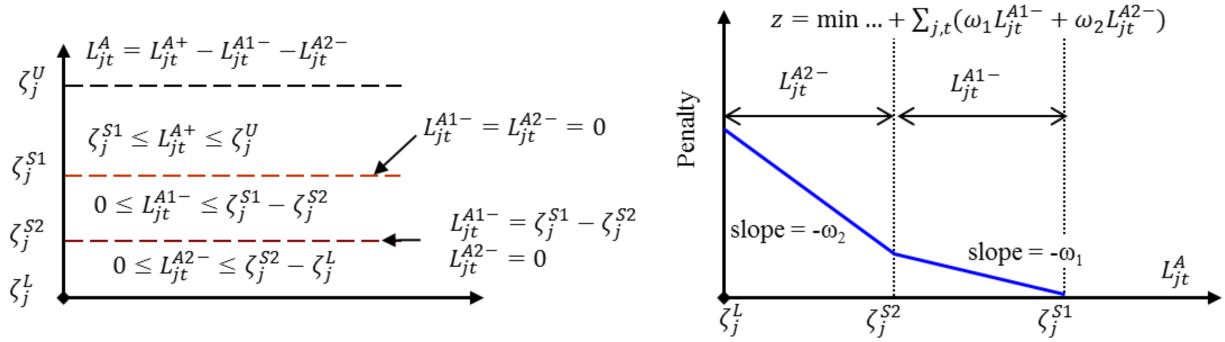


Figure 2.7. Piecewise linear penalties for inventory below ζ_j^{S1} and ζ_j^{S2}.

where $L_{jt}^{A+} \ge \zeta_j^S$ and $0 \le L_{jt}^{A-} \le \zeta_j^S - \zeta_j^L$. Together with Eq (2.7), (2.24) ensures the inventory between lower (ζ_j^L) and upper (ζ_j^U) bound, while the safety level violation L_{jt}^{A-} is penalized in the objective. Piecewise linear penalties can also be easily modeled, as shown in Figure 2.7.

2.4.2. Variable Loading/ Delivering Time

If there is a pumping rate for the truck (at most π_i units of product can be transferred in one period), the loading/delivering time is no longer fixed. To model this aspect we introduce the following binary and continuous variables:

(a) Delivering: $\bar{X}_{ijt}^D \in \{0,1\}$ is one if truck *i* is delivering at customer *j* during time period *t*.

(b) Loading: $\bar{X}_{it}^L \in \{0,1\}$ is one if truck *i* is being loaded at the plant during time period *t*.

(c) Delivery flow: $\bar{F}_{ijt}^D \ge 0$ is the delivery amount from truck *i* to customer *j* during period *t*.

(d) Load flow: $\bar{F}_{it}^L \ge 0$ is the loading amount to truck *i* during period *t*.

(e) Truck inventory: $L_{it}^T \ge 0$ is the inventory of truck *i* at time point *t*.

In the basic model, the delivering amount was equal to the difference of incoming and outgoing arc flows. Now the loading/delivering amount is modeled as follows,

$$\bar{X}_{ijt}^{D} \leq \bar{X}_{ijt}, \;\; \forall i, j \in \mathbf{J}^{C}, t \tag{2.25}$$

$$\bar{F}_{ijt}^{D} \leq \pi_i \bar{X}_{ijt}^{D}, \;\; \forall i, j \in \mathbf{J}^{C}, t \tag{2.26}$$

$$\bar{X}_{it}^{L} \leq \bar{X}_{iPt}, \;\; \forall i, t \tag{2.27}$$

$$\bar{F}_{it}^{L} \leq \pi_i \bar{X}_{it}^{L}, \;\; \forall i, t \tag{2.28}$$

Eq (2.25) ensures that a truck can deliver product to customer during a time period, if the truck is there, and Eq (2.26) enforces a maximum delivery amount per period. Similarly, Eq (2.27), (2.28) are written for the loading counterpart, following the same logic. Eq (2.25)-(2.28) replace Eq (2.4)-(2.5).

Also, the truck inventory is monitored as shown in the following equation,

$$L_{it}^{T} = L_{i,t-1}^{T} + \bar{F}_{it}^{L} - \sum_{j \in \mathbf{J}^{\mathbf{C}}} \bar{F}_{ijt}^{D} \leq \gamma_i, \;\; \forall i, t \tag{2.29}$$

For anticipatable customers, the inventory level L_{jt}^{A} decreases due to consumption, increases when there is a delivery flow, and is lower and upper bounded, as follows,

$$L_{jt}^{A} = L_{j,t-1}^{A} + \sum_{i} \bar{F}_{ijt}^{D} - \rho_{jt}, \;\; \forall j \in \mathbf{J}^{A}, t \tag{2.30}$$

$$\zeta_j^{L} \leq L_{jt}^{A} \leq \zeta_j^{U}, \;\; \forall j \in \mathbf{J}^{A}, t \tag{2.31}$$

For order-only customers, the delivery flow throughout the planning horizon should satisfy the ordered amount:

$$\sum_{i,t} \bar{F}_{ijt}^{D} \geq \varphi_j, \;\; \forall j \in \mathbf{J}^{O} \tag{2.32}$$

Thus, to model the variable loading/delivering time, Eq (2.6)-(2.8) shown in section 2.2 will be replaced by Eq (2.29)–(2.32).

2.4.3. Differentiation of Driving from Working

In addition to the maximum working time, there could be a maximum driving time limit, which requires that a driver cannot drive more than $\bar{\theta}^D$ hours (θ^D periods) cumulatively, without having a break of at least $\bar{\psi}$ hours (ψ periods).

Thus, keeping track of driving time is needed. We introduce a new binary variable $\bar{Y}_{ikt}^D$ which is one if driver k is driving truck i during time period t. The old variable $\bar{Y}_{ikt}^W$ is modified to be one if driver k is working with, but not driving truck i during time period t. The following formulation is based on the extension discussed in section 2.4.2 of variable loading/delivering time, with fixed loading/delivering time being zero.

Eq (2.18) is modified, so the engagement of a driver to a truck can be classified as driving, working but not driving, and resting:

$$\bar{Y}_{ikt} = \bar{Y}_{ikt}^D + \bar{Y}_{ikt}^W + \bar{Y}_{ikt}^R, \ \ \forall i,k,t \tag{2.33}$$

Next, the driving and working binary variables are constrained as follows:

$$\sum_k \bar{Y}_{ikt}^D = \sum_{j,j'} \sum_{t'=t-\tau_{jj'}}^{t-1} W_{ijj't'}, \ \ \forall i,t \tag{2.34}$$

$$\sum_k \bar{Y}_{ikt}^W \geq \bar{X}_{it}^L + \sum_{j \in \mathbf{J}^\mathbf{A}} \bar{X}_{ijt} + \sum_k \sum_{t'=t-\varphi^{CI}}^{t-1} Y_{ikt'}^S + \sum_k \sum_{t'=t}^{t+\varphi^{CO}-1} Y_{ikt'}^F, \ \ \forall i,t \tag{2.35}$$

Eq (2.34) requires that if the truck is being driven, then there will be a driver assigned to it and driving it. Eq 35 means that a driver is working, if the assigned truck is either: (1) loading product, (2) at a customer site, (3) checking in, or (4) checking out. The inequality in Eq (2.35) is used, again, to consider the case when the driver assigned to a truck is idle at the plant.

The maximum working/driving time constraints are written as follows

$$\sum_i \sum_{t' \in \mathbf{T}_t^{24}} (\bar{Y}_{ikt}^D + \bar{Y}_{ikt'}^W) \leq \theta^W, \;\; \forall k, t \tag{2.36}$$

$$\sum_i \sum_{t' \in \mathbf{T}_t^{21}} \bar{Y}_{ikt}^D \leq \theta^D, \;\; \forall k, t \tag{2.37}$$

Eq (2.36) is modified from Eq (2.23), based on the new definition of variable $\bar{Y}_{ikt}^W$, while Eq (2.37) means that a driver cannot drive more than θ^D periods cumulatively, where $\mathbf{T}_t^{21} = \{t' : t - (\bar{\psi} + \bar{\theta}^D)/\delta < t' \leq t\}$. Typically, $\bar{\theta}^D = 11$, so $\bar{\psi} + \bar{\theta}^D = 21$. To sum up, Eq (2.18), (2.20), (2.23) in **M2** will be replaced by Eq (2.33)-(2.37), when maximum driving time constraints need to be taken into account.

2.4.4. Drivers at the Plant

Practically, drivers can have more flexibility at the plant. For instance, a driver can drive two different trucks in one shift, thus, a checking-in (checking-out) does not necessarily coincide with the start (end) of a driver's day. Also, after a checking-out, a driver may wait for some time at the plant, idle but *on-the-clock*, before checking in with another truck. To model all the different situations arising when a driver is at the plant, the following three variables are introduced (see Figure 2.8).

(a) Idle: $\bar{V}_{kt}^P \in \{0,1\}$ is one if driver k is idle at the plant (not engaged with truck) during period t.

(b) Go to work: $U_{kt}^W \in \{0,1\}$ is one if driver k starts working at time point t.

(c) Go to rest: $U_{kt}^R \in \{0,1\}$ is one if driver k goes to rest at time point t.

With the new variables, the following constraints are introduced,

$$\bar{V}_{kt}^R = \bar{V}_{k,t-1}^R + U_{k,t-\psi-1}^R - U_{k,t-1}^W, \;\; \forall k, t \tag{2.38}$$

$$\bar{V}_{kt}^P = \bar{V}_{k,t-1}^P + \sum_i Y_{ik,t-1}^F + U_{k,t-1}^W - \sum_i Y_{ik,t-1}^S - U_{k,t-1}^R, \;\; \forall k, t \tag{2.39}$$

$$U_{kt}^W + U_{kt}^R \leq 1, \;\; \forall k, t \tag{2.40}$$

With the new variables, the following constraints are introduced,

Eq (2.38) is based on Eq (2.22), but requires that driver k is resting (beyond the minimum time) during period t, if: he has being resting or just finished the minimum resting hours; and did not go to work at the start of the current period. Eq (2.39) requires that a driver is idle at the plant if: (1) he was idle ($\bar{V}^P_{k,t-1} = 1$), or just checked out ($\sum_i Y^F_{ik,t-1} = 1$), or started working ($U^W_{k,t-1} = 1$); and (2) he did not check in ($\sum_i Y^S_{ik,t-1} = 0$) nor started resting ($U^R_{k,t-1} = 0$). Finally, Eq (2.40) enforces that driver cannot go to work and go to rest at the same time. To sum up, Eq (2.22) in **M2** will be replaced by Eq (2.38)-(2.40), when the different situations at the plant need to be modeled.

2.4.5. Remarks

The model can be easily extended to account for a wide range of additional restrictions. For example, we can add constraints to forbid the simultaneous deliveries of material from two trucks to the same customer by requiring that the summation of $\bar{X}^D_{ijt}$ over index *i* be less than or equal to 1, for each customer in each time period. Also, we can forbid the transfer of material from a customer to a truck. This can be accomplished by a constraint which requires that the sum of incoming flows be greater than the sum of outgoing flows, for each customer and time t. Although loading from a customer will in general be suboptimal, since this would incur additional traveling and set-up cost, there may be situations where loading from a customer with high inventory level can reduce the total cost. In the extension described in section 2.4.2, transferring from a customer to a truck is intrinsically infeasible, since $\bar{F}^D_{ijt}$ variable is non-negative.

Furthermore, the objective function can be modified to include driver cost explicitly, which may include the following terms: (1) the working time based wage ($\sum_{i,k,t} \omega^D_k \bar{Y}^W_{ikt}$ with ω^D_k being the hourly wage); (2) the shift-taking based wage ($\sum_{i,k,t} \omega^S_k Y^S_{ikt}$ with ω^S_k being the wage for every shift); and (3) the resting cost ($\sum_{i,j,t} \omega^R_{ij} W^R_{ijt}$ with ω^R_{ij} being the resting cost for truck *i* at customer *j*). Note that the resting cost can also be represented as $\sum_{i,k,t} \omega_k^R \bar{Y}_{ikt}^R$ where ω_k^R is the hourly resting-away cost for the driver.

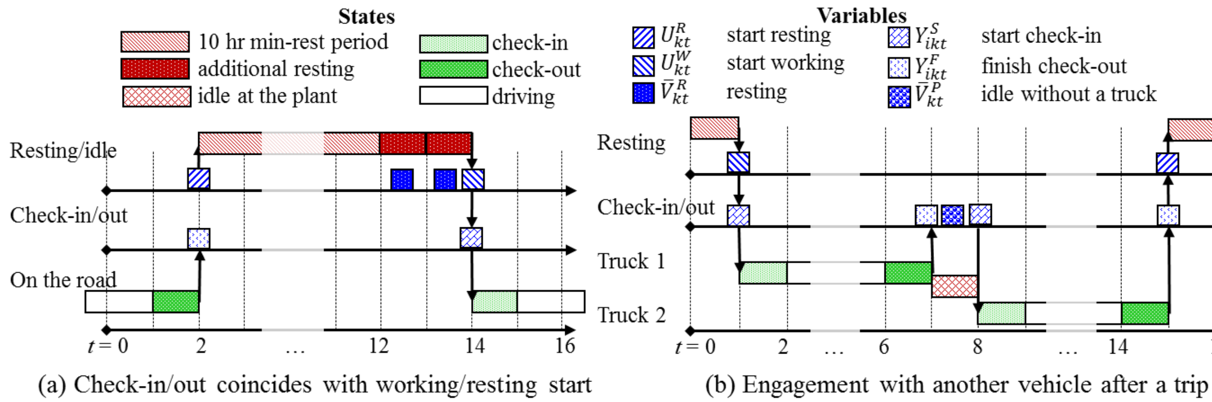


Figure 2.8. Driver modeling at the plant. (a) shows a case of going to rest, resting for more than the minimum period, and then checking in a truck. (b) shows a case of different truck engagements in one single shift, and the driver is idle without any truck for one period in this example.

Finally, note that solutions from **M1** or **M2** can be used to obtain initial solutions to models with the extensions discussed in section 2.4. Thus, it is possible to decompose the entire problem to different levels of detail and generate solutions of increasing complexity sequentially.

2.5. Example

A simple industrially inspired example is presented to illustrate what results can be obtained using the proposed model. There are 4 trucks and 5 drivers, which are always available, serving 8 customers. The objective is to minimize the distribution cost. The planning horizon is 36 hours, and the discrete time period is 1 hour. The check-in and check-out time is assumed to be 0.5 hour, the maximum daily working time is 14 hours, and the minimum resting time is 10 hours. The MIP model was implemented in GAMS 24.1 and solved using CPLEX 12.5 on a desktop with 3.4GHz Intel Core processor (i7-2600) and 8GB RAM, running Windows 7. The resource limit is 1,800s.

Due to confidentiality issues, inventory levels and capacities are expressed in terms of *material units*, denoted by MU. The capacities of truck T1, T2, T3, and T4 are 370, 383, 370, 374 MUs, respectively. The minimum loading parameter ε is 0.5. All trucks and drivers are initially located at the plant with no product loaded in trucks, and the trucks are required to return to the plant at the end of the horizon. The original travel times based on distance, without rounding or loading/delivering time, are shown in Figure 2.9(a). The fixed loading/delivering time is assumed to be 1 hour. Customers C1-C7 are anticipatable customers, while C8 is an order-only customer. The order window for C8 is from 24 to 36 hour point, and the order amount is 267 MUs. For simplicity, the consumption rates for anticipatable customers are assumed to be uniform throughout the horizon (given in Table 2.1). Other anticipatable customer data are also given in Table 2.1.

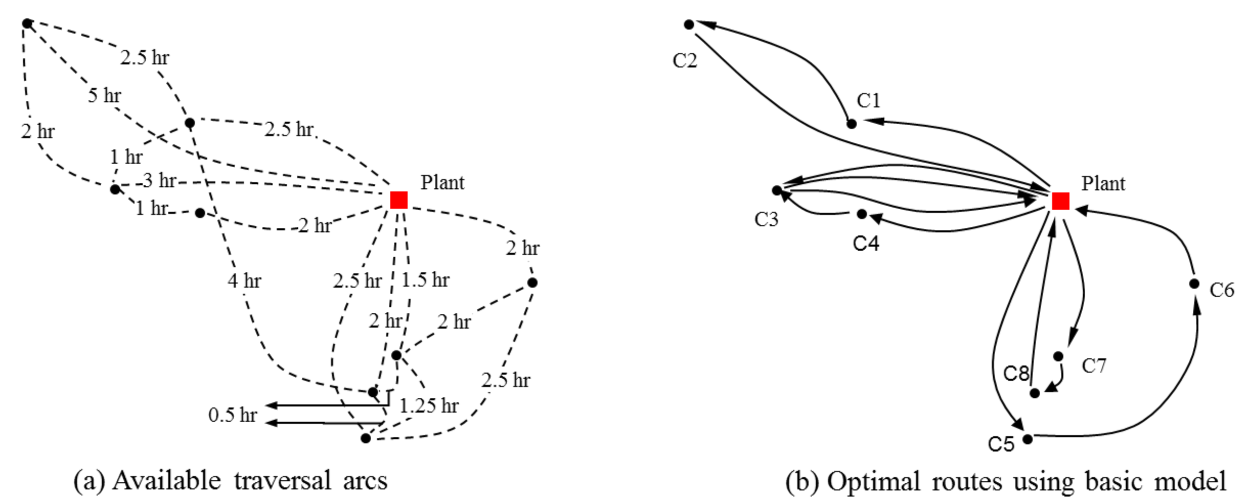


Figure 2.9. The 8-customer example. (a) Availability and original travel time of arcs. (b) Truck routes in the solution obtained using the basic model **M1**.

Table 2.1. Data for anticipatable customers; symbols and units in parentheses.

	C1	C2	C3	C4	C5	C6	C7
start/end time ($\sigma_{jm}^{AHS}/\sigma_{jm}^{AHE}$, hr)	0/36	0/36	0/36	0/36	0/36	0/24	0/24
consumption rate (ρ_{jt}, MUs/hr)	8.43	6.21	19.75	2.69	26.71	6.79	2.87
min/max level (ζ_j^L/ζ_j^U, MUs)	0/775	0/1513	0/815	0/609	0/4505	0/589	0/743
terminal minimum level (ζ_j^T, MUs)	488	1324	359	165	3554	193	111
initial inventory ($\hat{L}_{j0}^A$, MUs)	748	1375	376	239	4301	397	189

In the objective function, the distribution cost has three parts: (1) fuel cost, based on traveling distance, assuming that each hour of driving costs 4 cost units ($4 \cdot \sum_{i,j,j',t} \overline{\tau}_{jj'} W_{ijj't}$); (2) fixed delivery cost, assuming that each delivery costs 1 cost unit, $(\sum_{i,j,j' \in \mathbf{J}^{\mathrm{C}},t} W_{ijj't})$; and (3) resting away cost, which is equal to 5 cost units, $(5 \cdot \sum_{i,j \in \mathbf{J}^{\mathrm{C}},t} W_{ijt}^{R})$.

The basic model **M1** has 3,952 binary variables, 3,327 continuous variables, and 5,209 constraints. After 30 minutes, it yields a solution with an objective function value of 141 and an optimality gap equal to 16.18%. Figure 2.9(b) shows the truck routes of the solution obtained using **M1**. The solution with an objective value of 141 is actually optimal for this case.

When the valid inequalities in section 2.2.5 are removed from **M1**, its LP-relaxation decreases from 117.69 to 93.63, which means that the integrality gap increases from 16.5% to 33.6%. Also, while the model without the tightening constraints also obtains the optimal integer solution within 30 minutes, the optimality gap in this case is 31.63% instead of 16.18%. This shows that the cuts presented are effective in tightening the LP feasible region.

When driver constraints are considered with model **M2**, the number of binary, continuous variables, and constraints increases to 7,345 4,815 and 7,600 respectively. After 30 minutes, a solution with an objective function value of 146 is found with a 19.31% optimality gap. The solution is shown in Figure 2.10. The solutions of models **M1** and **M2** happen to include the same customers in the generated routes. However, in the solution of model **M2** a few visits got swapped, and driver K2 is resting at C1, to satisfy the maximum working time restriction. When the extensions described in section 5.2-4 are included, the model becomes computationally more expensive. The best solution found within 30 minutes has an objective function value of 151. A better solution, with an objective function equal to 146 and the same routes as **M1**, is found after 48 minutes.

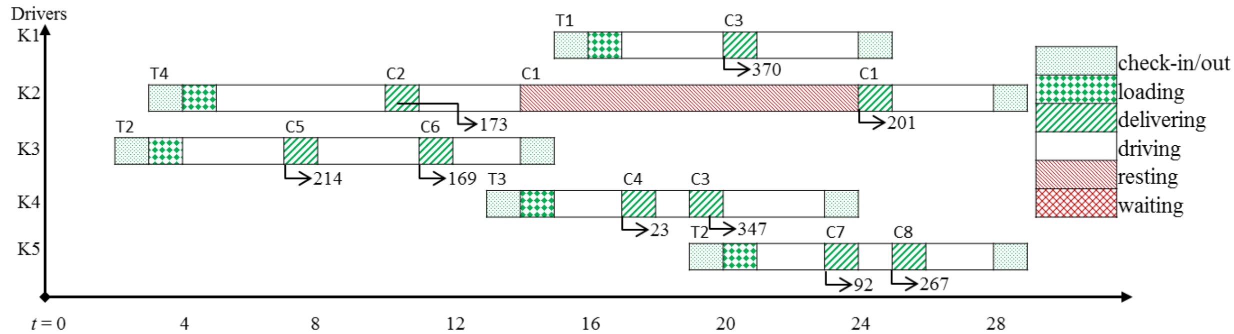


Figure 2.10. Gantt chart showing the solution using model **M2**.

2.6. Conclusions

One of the major challenges in the adoption of optimization-based methods for inventory routing problems is the wide range of constraints that an IR solution should satisfy in order to be implementable. To our knowledge, no systematic optimization-based method to address the general IRP is currently available for truck-based distribution networks. Towards this challenge, we developed a MIP framework for IR in industrial gases supply chains. Our framework allows us to formulate models that account for a number of complex features simultaneously, including maximum daily working and driving time per driver; time-varying consumption rates; customer access hours; and heterogeneous fleet. We also showed how the framework allows us to consider additional features. Most importantly, our analysis shows that the solutions found using the proposed MIP models within a reasonable time, are on average better than the solutions manually generated by the logistic planners based on heuristic rules.

Nevertheless, the proposed model becomes prohibitively expensive for larger instances, so the development of advanced solution methods is necessary. The proposed framework can also be used as the basis for the formulation of more effective MIP models as well as the design of decomposition methods. For example, the basic model can be used for the generation of routes that can then be used as input to a second level optimization model (employing discrete or continuous modeling of time) for the assignment of drivers to routes subject to driving, working, and resting restrictions.

2.7. Notation

Indices/Sets

$i \in \mathbf{I}$	trucks
$j \in \mathbf{J}$	supply chain nodes
$k \in \mathbf{K}$	drivers
$m \in \mathbf{M}_j^{AH}$	access window of customer *j*
$t \in \mathbf{T}$	time periods or points

Subsets

$\mathbf{J}^{\mathbf{C}}$	customers
$\mathbf{J}^{\mathbf{A}}$	anticipatable customers
$\mathbf{J}^{\mathbf{O}}$	order-only customers
$\mathbf{T}_t^{21}/\mathbf{T}_t^{24}$	time periods included in the 21/24-hour interval ending with period *t*

Binary Variables

U_{kt}^W/U_{kt}^R	=1 if driver *k* starts working/resting at time point *t*
$\bar{V}_{kt}^P$	=1 driver *k* is idle at the plant (not engaged with any truck) during period *t*
$\bar{V}_{kt}^R$	=1 if driver *k* is resting at the plant beyond the minimum ψ periods, during period *t*
$W_{ijj't}$	= 1 if truck *i* starts trip from *j* to *j'* at time point *t*
W_{ijt}^R	=1 if the driver assigned to truck *i* starts a rest period after visiting customer *j*, at time point *t*
$\bar{X}_{ijt}$	=1 if truck *i* is at SC node *j* during time period *t*
$\bar{X}_{ijt}^D$	=1 if truck *i* is delivering at customer *j* during time period *t*
$\bar{X}_{it}^L$	=1 if truck *i* is being loaded at the plant during time period *t*
$\bar{Y}_{ikt}$	=1 if driver *k* is assigned to truck *i* during time period *t*
Y_{ikt}^S/Y_{ikt}^F	=1 if driver *k* starts to check-in/ finishes check-out of truck *i* at time point *t*

$\bar{Y}_{ikt}^{W}/\bar{Y}_{ikt}^{D}/\bar{Y}_{ikt}^{R}$ =1 if driver k is working/ driving/ resting away from the plant, with truck i during time period t

Non-Negative Variables

$F_{ijj't}^{A}$	product loaded in truck *i*, which starts the trip from node *j* to node *j'* at time *t*
$\bar{F}_{ijt}^{D}$	delivery amount from truck *i* to customer *j* during period *t*
$\bar{F}_{it}^{L}$	loading amount to truck *i* during period *t*
L_{jt}^{A}	inventory level of anticipatable customer *j* at time point *t*
L_{jt}^{A+}/L_{jt}^{A-}	inventory level above/below safety level (for penalization), of anticipatable customer *j* at time point *t*
L_{it}^{T}	inventory level of truck *i* at time point *t*

Parameters

$\alpha_{jt}^{AH}/\alpha_{jt}^{O}$	=1 if period *t* is within one of the accessible/ order window of customer *j*
β_j	fixed loading/delivering time at SC node *j*
γ_i	capacity of truck *i*
δ	time period length
ε	minimum loading percentage
$\zeta_j^{L}/\zeta_j^{U}/\zeta_j^{S}/\zeta_j^{T}$	minimum/ maximum/ safety/ terminal minimum level of anticipatable customer *j*
η	planning horizon
θ^{W}/θ^{D}	maximum working/driving time without resting, in unit of time periods
π_i	pumping rate of truck *i* in one time period
ρ_{jt}	consumption amount for anticipatable customer *j* during period *t*
$\sigma_{jm}^{AHS}/\sigma_{jm}^{AHE}$	start/end time of access window *m* of customer *j*
$\sigma_j^{OS}/\sigma_j^{OE}$	start/end time of order window of customer *j*
$\tau_{jj'}$	travel time of arc (*j*,*j'*), in unit of time periods

φ_j	order amount of order-only customer *j*
$\varphi^{CI}/\varphi^{CO}$	check-in/check-out time, in units of time periods
ψ	minimum resting time, in unit of time periods
$\omega_{ijj'}$	travel cost of arc (*j,j'*) for truck *i*
ω_{ij}^{R}	assigned driver resting cost for truck *i* at customer *j*
$\omega_k^D/\omega_k^S/\omega_k^R$	driving/shift taking/resting payment for driver *k*
$\hat{F}_{ijj't}^{A}$	the truck inventory, if truck *i* is travelling on the arc (*j,j'*) starting at *t*, zero otherwise
$\hat{L}_{j0}^{A}$	initial inventory of anticipatable customer *j*
$\hat{L}_{i0}^{T}$	initial inventory of truck *i*
$\widehat{W}_{ijj't}$	=1 if truck *i* is on the road, due to pre-assigned trip from *j* to *j'* starting at time point *t*
$\hat{X}_{ij0}$	=1 if initial location of truck *i* is *j*

Chapter 3

Solution Methods for IRP under VMI Policy with Driver Constraints[3]

A MIP model for IRP that addresses all the complex constraints has been proposed in the previous chapter, but it becomes intractable for large instances. Accordingly, the goal of this chapter 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 chapter is structured as follows. In Section 3.1, we provide a detailed problem statement, and summarize the solution methods. In Section 3.2, 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 3.3, 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 3.4, 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 3.5, we present an iterative approach that integrates the two subproblems. In Section 3.6, different instances are presented. 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.

3.1. Problem and Method Overview

3.1.1. Problem Statement

[3] This chapter is modified from Dong et al., 2017.

The problem is represented by the following: a set of trucks, $i \in \mathbf{I}$; a set of SC nodes, $j \in \mathbf{J}$, which includes a central plant P, and a subset of customers $\mathbf{J}^C$; and a set of drivers, $k \in \mathbf{K}$. The objective is to find the optimal delivery amounts, routes, schedules, and resource allocations (drivers, trucks), to minimize the distribution cost, subject to the constraints described below. We assume that there is only one central plant, and the liquid gases are always available at the plant. It is also assumed that there is only one product in the problem, as different products use different trailers and are scheduled independently.

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 §3.4, 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_j^U, of the tank and the minimum inventory level, ζ_j^L. 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_j^{OS} and σ_j^{OE}, 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.

3.1.2. Solution Strategy

The proposed solution strategy includes three components, described in §3.2-§3.4. 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.

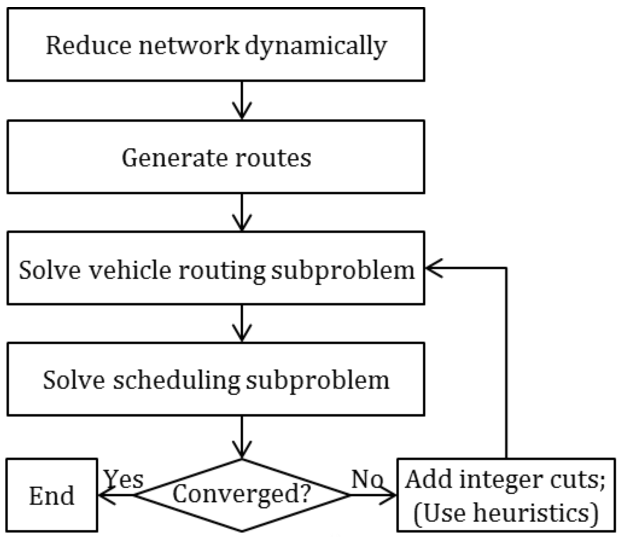


Figure 3.1. Outline of the solution strategy.

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 §3.5. A simplified flowchart of the solution approach is shown in Figure 3.1.

3.2. 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.

3.2.1. 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 $\rho_j^T(t)$ denote the time-varying consumption rate of customer j, and $L0_j^A$ denote its initial inventory. The *minimum* and *maximum demand* for each customer can be calculated as follows:

$$\sigma_j^{\text{MIN}} = \begin{cases} \max(0, \zeta_j^S + \int_0^\eta \rho_j^T(t)dt - L0_j^A) & \text{if } j \in \mathbf{J}^A \\ \varphi_j & \text{if } j \in \mathbf{J}^O \end{cases} \tag{3.1}$$

$$\sigma_j^{\text{MAX}} = \begin{cases} \zeta_j^U + \int_0^\eta \rho_j^T(t)dt - L0_j^A & \text{if } j \in \mathbf{J}^A \\ \varphi_j & \text{if } j \in \mathbf{J}^O \end{cases} \tag{3.2}$$

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., $\mathbf{J}^T = \{j | \sigma_j^{\text{MIN}} > 0\}$. This idea is illustrated in Figure 3.2.

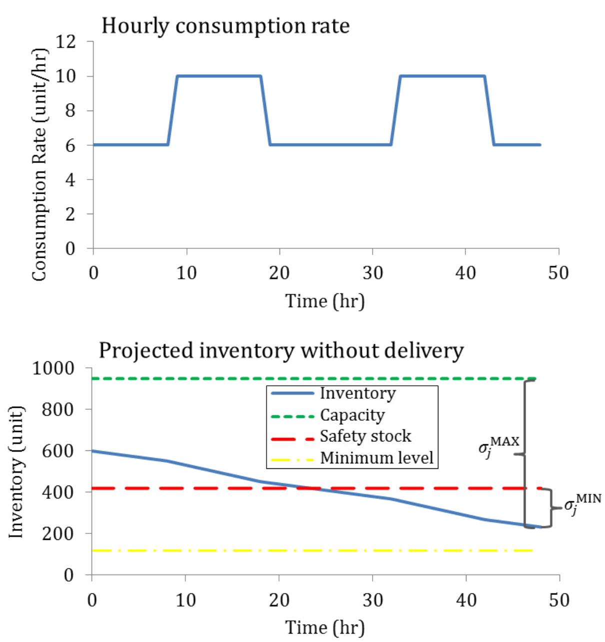


Figure 3.2. The procedure of determining trigger customers.

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

$$\zeta_j^S = \max\{a \cdot \zeta_j^U, \zeta_j^L + \bar{\tau}_{Pj} \cdot \bar{\rho}_j + b \cdot \sqrt{\bar{\tau}_{Pj} \cdot \delta^2(\rho_j) + \bar{\rho}_j^{\,2} \cdot \delta^2(\tau_{Pj})}\} \tag{3.3}$$

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: $\bar{\tau}_{Pj}$ /$\bar{\rho}_j$ 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 $\bar{\tau}_{Pj} \cdot \bar{\rho}_j$ 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 equation (3.3) 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 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 Figure 3.3). The adjustable parameters defining this region are the angle θ, and the radius r. When θ =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*.

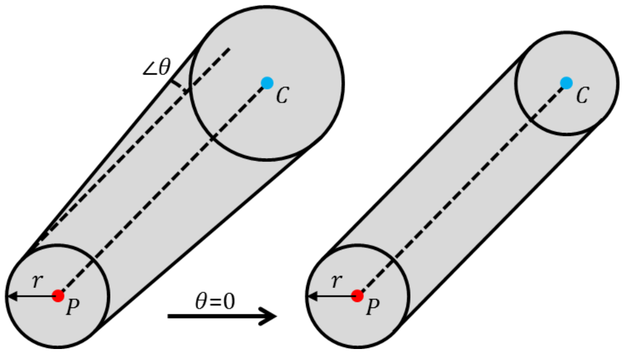


Figure 3.3. Illustration of the trigger customer region. C is the trigger customer, and P is the plant.

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 (1) consumption in the planning horizon, (2) the consumption in T_j days following the current horizon, and (3) 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 Figure 3.4(a)). 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,

$$T_j = \max\{\overline{T} - \left\lfloor \frac{C_j/r^2}{\bar{C}/\overline{r}^2} \right\rfloor, 1\} \tag{3.4}$$

where $\overline{T}$ is the user-defined largest possible T_j, $\overline{r}$ is the maximum distance between any customer and the plant, $\bar{C}$ 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 hours), C_j is the number of other customers within the disk of radius r around customer j. With T_j defined in equation (3.4), which is illustrated in Figure 3.4(b), 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.

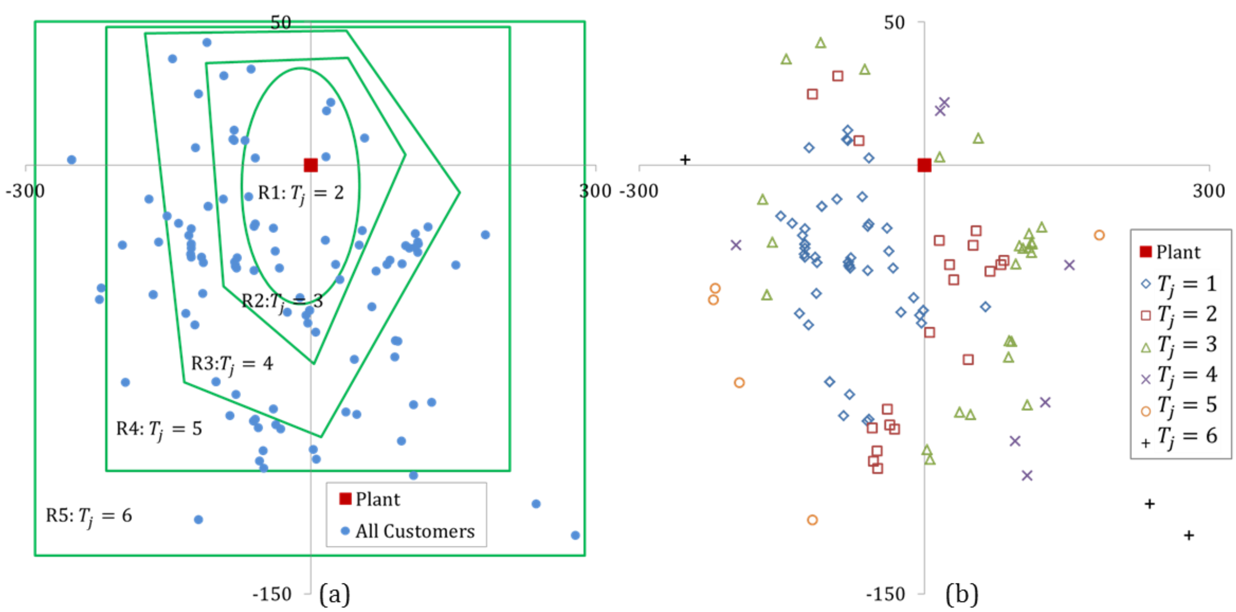


Figure 3.4. Illustration of different T_j definition in inventory level criterion, with both axes in unit of miles. (a) is for option A to consider plant-customer distance, in which customers are divided into regions R1-R5. (b) is for option B to consider customer density.

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

$$\mathbf{J}^B = \left\{ j' \middle| j' \in \bigcup_{j \in \mathbf{J}^T} \mathbf{J}_j^R \quad \text{and} \quad L0_{j'}^A - \int_0^{\eta + 24T_{j'}} \rho_{j'}^T(t)dt < \zeta_{j'}^S \right\} \tag{3.5}$$

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' that is within the trigger customer region and has the greatest $\sigma_{j'}^{\text{MAX}}$ 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.

3.2.2. 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,

$$\tau o_{j,j'} \geq \max[c \cdot \theta^D, d \cdot (\tau o_{j,P} + \tau o_{j',P})] \tag{3.6}$$

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 (3.6) requires that the travel time from *j* to *j'* is greater than both (1) a percentage of the maximum daily driving time and (2) a percentage of the travel time of *j*->*P*->*j'*. Typically, *c* and *d* are selected between 0.7-1. 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 G.

3.2.3. Example

The customer set shown in Figure 3.4 is used as an example. The planning horizon is 2 days. Parameter T_j is based on Figure 3.4(a), and the trigger customer region is defined using option A (θ=10° and *r*=10 miles). The preprocessing algorithm identifies 18 trigger customers, and 14 balance customers (see Figure 3.5). The number of customers drops from 111 to 32, and the number of directed arcs drops from 6067 to 485.

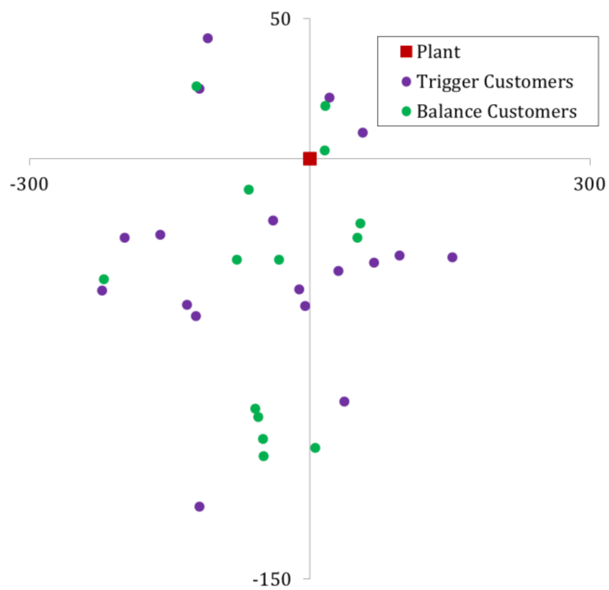


Figure 3.5. SC nodes in the distribution network after dynamic network reduction.

3.3. 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.

3.3.1. 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,

$$\tau_r^D = \sum_{(j,j')\in\mathbf{A}_r} \tau o_{j,j'} \tag{3.7}$$

$$\tau_r^W = \sum_{(j,j')\in\mathbf{A}_r} \tau_{j,j'} + \sum_{(j,j')\in\mathbf{A}_r:j,j'\neq P} \max(0, \min_m \sigma_{j',m}^{AS} - \max_m \sigma_{j,m}^{AE} - \tau o_{j,j'}) \tag{3.8}$$

$$\tau_r^R = \begin{cases} \tau_r^W & \text{if } \tau_r^D \le \theta^D \text{ and } \tau_r^W \le \theta^W \\ \tau_r^W + \psi & \text{otherwise} \end{cases} \tag{3.9}$$

$$\gamma_r^R = \begin{cases} \gamma^D \cdot \tau_r^D + \gamma^W \cdot \tau_r^W + \gamma^V \cdot |\mathbf{J}_r| & \text{if } \tau_r^D \le \theta^D \text{ and } \tau_r^W \le \theta^W \\ \gamma^D \cdot \tau_r^D + \gamma^W \cdot \tau_r^W + \gamma^V \cdot |\mathbf{J}_r| + \gamma^R & \text{otherwise} \end{cases} \tag{3.10}$$

Each route in the generated route set **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->j'->j"*, is included in **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->j'* is included in **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 $\max_m \sigma_{j,m}^{AE} \geq \sum_{(j',j'') \in \mathbf{A}_{r,j}^{RP}} \tau_{j',j''}$, 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., $\tau_r^W \leq 2\theta^W$ and $\tau_r^D \leq 2\theta^D$.

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

3.3.2. 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) $F^{R}_{i,r,j} \geq 0$: delivery amount from truck *i* to customer *j* using route *r*.

(c) $F^{RX}_{i,r} \geq 0$: 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,

$$\min O^{VR} = \sum_{i,r} (\gamma^{R}_{r} Z_{i,r} + \gamma^{X} F^{RX}_{i,r}) \tag{3.11}$$

$$\sum_{j \in \mathbf{J}_r} F^{R}_{i,r,j} + F^{RX}_{i,r} = \xi_i Z_{i,r}, \quad \forall i, r \tag{3.12}$$

$$F^{R}_{i,r,j} \leq (\zeta^{U}_{j} - \zeta^{L}_{j}) Z_{i,r}, \quad \forall i, r, j \in \mathbf{J}^{A} \cap \mathbf{J}_r \tag{3.13}$$

$$\sigma^{\text{MIN}}_{j} \leq \sum_{i,r \in \mathbf{R}_j} F^{R}_{i,r,j} \leq \sigma^{\text{MAX}}_{j}, \quad \forall j \in \mathbf{J}^{C} \tag{3.14}$$

$$\sum_{r} \tau^{R}_{r} Z_{i,r} \leq \eta, \quad \forall i \tag{3.15}$$

The objective function (3.11) accounts for the routing cost, and a penalty term for unused truck capacity (γ^{X} per unit of material). Constraints (3.12) enforce the truck capacity, and fix the delivery amounts to zero if route *r* is not used by truck *i*. Constraints (3.13) enforce that each delivery cannot exceed the difference between the maximum and minimum inventory levels, while constraints (3.14) enforce demand satisfaction for each customer. Constraints (3.15) 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,

$$\sum_{i,r\in\mathbf{R}_j:\alpha\tau_{r,j}\leq\omega\tau_j} Z_{i,r} \geq 1, \quad \forall j \in \mathbf{J}^A \tag{3.16}$$

where $\omega\tau_j$ denote the time when the projected inventory of customer j (without delivery) goes below its lower bound (defined in equation (3.17) below), and $\alpha\tau_{r,j}$ denote the earliest possible time to visit j on route r (defined in equation (3.18) below). Thus, constraints (3.16) 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.

$$\omega\tau_j = \min_t\{t|L0_j^A - \int_0^t \rho_{j,t'}^T dt' \leq \zeta_j^L\} \tag{3.17}$$

$$\alpha\tau_{r,j} = \sum_{(j',j'')\in\mathbf{A}_{r,j}^{RP}} \tau_{j'j''} \tag{3.18}$$

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,

$$\sum_{i,r\in\mathbf{R}_j:|\mathbf{J}_r|=1} Z_{i,r} \geq 1, \quad \forall j \in \mathbf{J}^A: \sigma_j^{\mathrm{MIN}} \geq \max_i \xi_i \tag{3.19}$$

where $r \in \mathbf{R}_j: |\mathbf{J}_r| = 1$ is the single-customer route visiting j. Note that constraints (3.19) 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.

3.4. 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.

3.4.1. 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*.

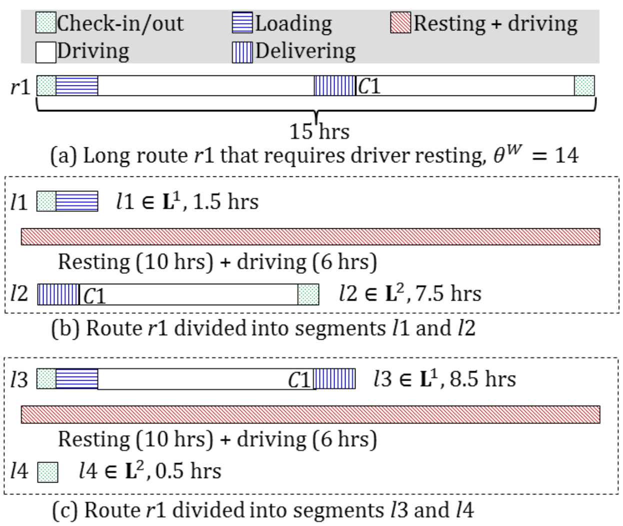


Figure 3.6. All ways to break long routes into segments are considered.

Throughout the chapter, 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 Figure 3.6.

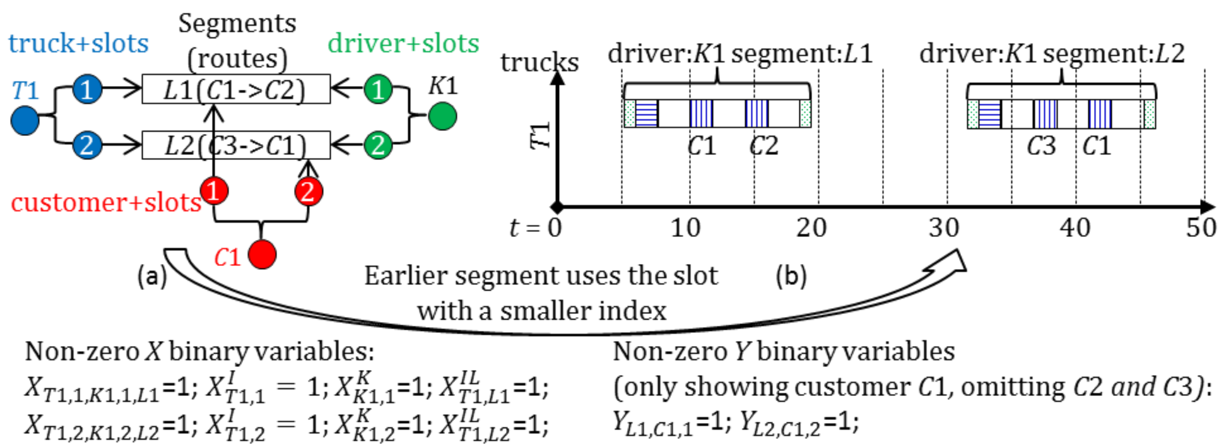


Figure 3.7. Illustration of slots and binary variables; two routes/segments are assigned to the same truck (T1) and driver (K1), and one customer (C1) appears in both routes.

Second, sets **R**, **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 $n \in \mathbf{N} = \{1, \dots, maxN\}$ 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 Figure 3.7(a)). Specifically, the following sets are defined:

(a) $\mathbf{N}^I = \{1, \dots, N^{maxi}\} \subseteq \mathbf{N}$: 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., $N^{maxi} = \max_i(\sum_r Z_{i,r})$.

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

(c) $\mathbf{N}^K = \{1, \dots, N^{maxk}\} \subseteq \mathbf{N}$: 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,

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

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}).

3.4.2. 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 Figure 3.7(b), 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.

3.4.3. Segment Assignment Constraints

Segments are assigned to different trucks and drivers as follows,

$$\sum_{k,n' \in \mathbf{N}^K, l \notin \mathbf{L}^2} X_{i,n,k,n',l} = X_{i,n}^I \quad \forall i,n \in \mathbf{N}^I \tag{3.20}$$

$$X_{i,n}^I \geq X_{i,n+1}^I \quad \forall i,n \in \mathbf{N}^I \tag{3.21}$$

$$\sum_{i,n \in \mathbf{N}^I, l} X_{i,n,k,n',l} = X_{k,n'}^K \quad \forall k,n' \in \mathbf{N}^K \tag{3.22}$$

$$X_{k,n}^K \geq X_{k,n+1}^K \quad \forall k,n \in \mathbf{N}^K \tag{3.23}$$

Constraints (3.20) define the truck aggregated variable $X_{i,n}^I$, while constraints (3.21) are used for symmetry breaking. Constraints (3.22) define the driver aggregated variable $X_{k,n}^K$, and constraints (3.23) break the symmetry in the same way as constraints (3.21). Note that the summation in constraints (3.20) excludes the second segment of long routes, $\mathbf{L}^2$, while constraints (3.22) 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.

$$\sum_{n\in \mathbf{N}_i^I, k, n'\in \mathbf{N}^K} X_{i,n,k,n',l} = X_{i,l}^{IL} \quad \forall i,l \tag{3.24}$$

$$\sum_{i,l\in \mathbf{L}_r \setminus \mathbf{L}^2} X_{i,l}^{IL} = \mu_r \quad \forall r \tag{3.25}$$

$$X_{i,n,k,n',l} = X_{i,n,k,n'+1,l'} \quad \forall i,n\in \mathbf{N}^I, k,n'\in \mathbf{N}^K, l\in \mathbf{L}^1, l'\in \mathbf{L}_l^{next} \tag{3.26}$$

Constraints (3.24) define the truck-segment aggregated variable $X_{i,l}^{IL}$, while constraints (3.25) 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 (3.26) 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* ($i \notin \mathbf{I}_l$).

3.4.4. 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.

$$S_{i,n,k,n',l,j} \le \eta \cdot X_{i,n,k,n',l} \quad \forall i,n\in \mathbf{N}^I, k,n'\in \mathbf{N}^K, l,j\in \mathbf{J}_l \tag{3.27}$$

$$E_{i,n,k,n',l,j} \le \eta \cdot X_{i,n,k,n',l} \quad \forall i,n\in \mathbf{N}^I, k,n'\in \mathbf{N}^K, l,j\in \mathbf{J}_l \tag{3.28}$$

$$S_{i,n}^I = \sum_{k,n'\in \mathbf{N}^K, l\notin \mathbf{L}^2, j\in \mathbf{J}_l^{start}} S_{i,n,k,n',l,j} \quad \forall i,n\in \mathbf{N}^I \tag{3.29}$$

$$E_{i,n}^I = \sum_{k,n'\in \mathbf{N}^K, l\notin \mathbf{L}^1, j\in \mathbf{J}_l^{end}} E_{i,n,k,n',l,j} \quad \forall i,n\in \mathbf{N}^I \tag{3.30}$$

$$E_{i,n}^{I} \le S_{i,n+1}^{I} + \eta \cdot (1 - X_{i,n+1}^{I}) \quad \forall i, n \in \mathbf{N}^{I} \tag{3.31}$$

Constraints (3.27)/(3.28) 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 (3.29)/(3.30) define the truck start/end time variables $S_{i,n}^{I}/E_{i,n}^{I}$, while constraints (3.31) state that slot *n*+1 of truck *i* cannot start before slot *n* of the same truck is finished.

$$S_{k,n'}^{K} = \sum_{i,n \in \mathbf{N}^{I}, j \in \mathbf{J}_{l}^{start}} S_{i,n,k,n',l,j} \quad \forall k, n' \in \mathbf{N}^{K} \tag{3.32}$$

$$E_{k,n'}^{K} = \sum_{i,n \in \mathbf{N}^{I}, j \in \mathbf{J}_{l}^{end}} E_{i,n,k,n',l,j} \quad \forall k, n' \in \mathbf{N}^{K} \tag{3.33}$$

Constraints (3.32)/(3.33) define the driver start/end time variables $S_{k,n}^{K}/E_{k,n}^{K}$. (The difference between them and constraints (3.29), (3.30) is due to the same reason as for $X_{k,n}^{K}$ and $X_{i,n}^{I}$). 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 $S_{k,n}^{K}/E_{k,n}^{K}$.

$$R_{k,n} \le X_{k,n}^{K} \quad \forall k, n \in \mathbf{N}^{K} \tag{3.34}$$

$$S_{k,n+1}^{K} - E_{k,n}^{K} \ge \psi \cdot R_{k,n+1} - \eta \cdot (1 - X_{k,n+1}^{K}) \quad \forall k, n \in \mathbf{N}^{K} \backslash \{N^{maxk}\} \tag{3.35}$$

$$E_{k,n+1}^{K} - S_{k,n}^{K} \le \theta^{W} + \eta \cdot R_{k,n+1} \quad \forall k, n \in \mathbf{N}^{K} \backslash \{N^{maxk}\} \tag{3.36}$$

Constraints (3.34)-(3.36) express restrictions on the working and resting time of drivers. Constraints (3.34) require $R_{k,n}$ to be zero if $X_{k,n}^{K}$ is zero. Constraints (3.35) 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 (3.36) 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}.

$$S^K_{k,n-1} + 2\theta^W + \psi \geq E^K_{k,n+1} - \eta \cdot (R_{k,n} + 1 - \sum_{i,n' \in \mathbf{N}^I, l \in \mathbf{L}^1} X_{i,n',k,n,l}) \quad \forall k, n \in \mathbf{N}^K \setminus \{1, N^{maxk}\} \tag{3.37}$$

$$S^K_{k,n-1} + 2\theta^W + \psi \geq E^K_{k,n+1} - \eta \cdot (R_{k,n+1} + 1 - \sum_{i,n' \in \mathbf{N}^I, l \in \mathbf{L}^2} X_{i,n',k,n,l}) \; \forall k, n \in \mathbf{N}^K \setminus \{1, N^{maxk}\} \tag{3.38}$$

Constraints (3.37) exclude schedules that have a long route succeeding a short route directly, and violate the working time limit, as depicted in Figure 3.8(a). 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 (3.38) follow the same idea, for the case of a short route succeeding a long route directly.

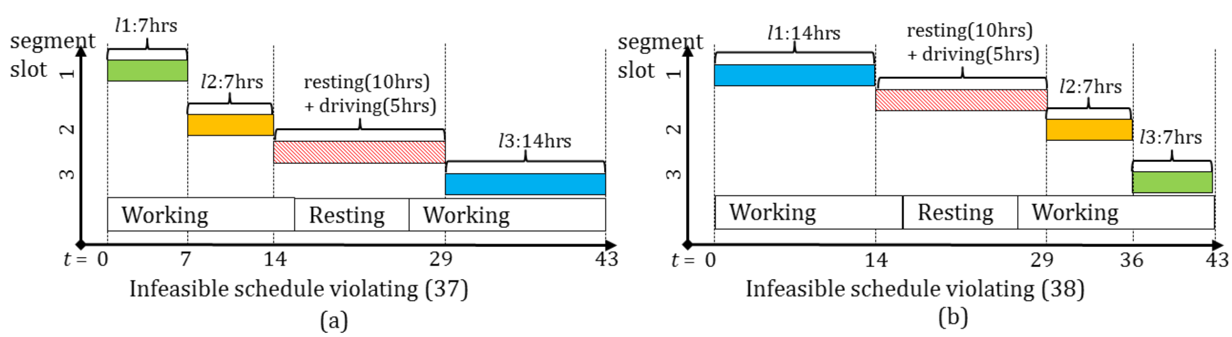


Figure 3.8. Illustration of infeasible schedules that are cut off by (3.37) and (3.38) . For both cases, the resting time limit is 10 hours, while the maximum daily working time limit is 14 hours.

$$E_{i,n,k,n',l,j} = S_{i,n,k,n',l,j} + \vartheta_j \cdot X_{i,n,k,n',l} + \omega_{i,j} \cdot F_{i,n,k,n',l,j} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \tag{3.39}$$

$$S_{i,n,k,n',l,j'} \geq E_{i,n,k,n',l,j} + \tau o_{j,j'} - \eta \cdot (1 - X_{i,n,k,n',l}) \; \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, (j,j') \in \mathbf{A}_l \tag{3.40}$$

$$\begin{gathered} S_{i,n,k,n'+1,l',j'} \geq E_{i,n,k,n',l,j} + \tau o_{j,j'} + \psi - \eta \cdot (1 - X_{i,n,k,n',l}) \\ \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l \in \mathbf{L}^1, l' \in \mathbf{L}^{next}_l, j \in \mathbf{J}^{end}_l, j' \in \mathbf{J}^{start}_{l'} \end{gathered} \tag{3.41}$$

Constraints (3.39) 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 (3.40) 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 (3.39), where $\omega_{i,j}$ is the reciprocal of the rate of delivery. Constraints (3.41) 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'*.

$$S_l^L = \sum_{i,n\in\mathbf{N}^I,k,n'\in\mathbf{N}^K,j\in\mathbf{J}_l^{start}} S_{i,n,k,n',l,j} \quad \forall l \tag{3.42}$$

$$E_l^L = \sum_{i,n\in\mathbf{N}^I,k,n'\in\mathbf{N}^K,j\in\mathbf{J}_l^{end}} E_{i,n,k,n',l,j} \quad \forall l \tag{3.43}$$

$$E_l^L \le S_l^L + \theta^W \quad \forall l \tag{3.44}$$

$$E_{l'}^L \le S_l^L + 2\theta^W + \psi \quad \forall l \in \mathbf{L}^1, l' \in \mathbf{L}_l^{next} \tag{3.45}$$

Constraints (3.42)/(3.43) define the segment start/end time variables S_l^L/E_l^L. Constraints (3.44) and (3.45) express restrictions on the durations of a single-route segment and a long route with two segments.

3.4.5. Delivery Flow Constraints

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

$$F_{i,n,k,n',l,j} \le \xi_i \cdot X_{i,n,k,n',l} \quad \forall i,n \in \mathbf{N}^I, k,n' \in \mathbf{N}^K, l,j \in \mathbf{J}_l \cap \mathbf{J}^C \tag{3.46}$$

$$F_{i,l}^{SX} + \sum_{n\in\mathbf{N}^I,k,n'\in\mathbf{N}^K,j\in\mathbf{J}_l\cap\mathbf{J}^C} F_{i,n,k,n',l,j} = \xi_i \cdot X_{i,l}^{IL} \quad \forall i, l \in \mathbf{L}^S \tag{3.47}$$

$$F_{i,l}^{SX} + \sum_{n\in\mathbf{N}^I,k,n'\in\mathbf{N}^K,j\in\mathbf{J}_l^L\cap\mathbf{J}^C} F_{i,n,k,n',l,j} + \sum_{n\in\mathbf{N}^I,k,n'\in\mathbf{N}^K,l'\in\mathbf{L}_l^{next},j'\in\mathbf{J}_{l'}\cap\mathbf{J}^C} F_{i,n,k,n',l',j'} = \xi_i \cdot X_{i,l}^{IL} \quad \forall i, l \in \mathbf{L}^1 \tag{3.48}$$

Constraints (3.46) enforce no product delivery when $X_{i,n,k,n',l} = 0$. Truck capacity constraints are expressed in constraints (3.47) and (3.48), respectively for short and long routes. In constraints (3.48), the two summations represent the delivery amount on the first and the second segments of a long route.

$$F_{l,j}^{LJ} = \sum_{i,n \in \mathbf{N}_i^I, k,n' \in \mathbf{N}_k^K} F_{i,n,k,n',l,j} \quad \forall l, j \in \mathbf{J}_l \tag{3.49}$$

$$\sigma_j^{\mathrm{MIN}} \le \sum_{l \in \mathbf{L}_j} F_{l,j}^{LJ} \le \sigma_j^{\mathrm{MAX}} \quad \forall j \in \mathbf{J}^C \tag{3.50}$$

Constraints (3.49) define the segment-customer aggregated delivery flow variable $F_{l,j}^{LJ}$. Constraints (3.50) state that the total delivery amount to a customer should satisfy its minimum and maximum demands.

3.4.6. Access Window Constraints

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

$$\sum_m W_{i,n,k,n',l,j,m} = X_{i,n,k,n',l} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \cap \mathbf{J}^C \tag{3.51}$$

$$S_{i,n,k,n',l,j} \ge \sum_m \sigma_{j,m}^{AS} \cdot W_{i,n,k,n',l,j,m} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \cap \mathbf{J}^C \tag{3.52}$$

$$E_{i,n,k,n',l,j} \le \sum_m \sigma_{j,m}^{AE} \cdot W_{i,n,k,n',l,j,m} \quad \forall i, n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K, l, j \in \mathbf{J}_l \cap \mathbf{J}^C \tag{3.53}$$

Constraints (3.51) 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 (3.52) and (3.53) enforce access window restrictions.

3.4.7. Inventory Constraints

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

$$\sum_{n \in \mathbf{N}_j^J} Y_{l,j,n} = \sum_i X_{i,l}^{IL} \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A \tag{3.54}$$

$$\sum_{l \in \mathbf{L}_j} Y_{l,j,n} = 1 \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}_j^J \tag{3.55}$$

Constraints (3.54) state that if a segment is carried out, the visit to an anticipatable customer corresponds to one of the customer slots. Constraints (3.55) require that every slot of an anticipatable customer corresponds to a segment that contains this customer.

$$S_{l,j}^{LJ} = \sum_{i,n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K} S_{i,n,k,n',l,j} \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A \tag{3.56}$$

$$E_{l,j}^{LJ} = \sum_{i,n \in \mathbf{N}^I, k, n' \in \mathbf{N}^K} E_{i,n,k,n',l,j} \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A \tag{3.57}$$

$$S_{l,j}^{LJ} - \eta \cdot (1 - Y_{l,j,n}) \leq S_{j,n}^{JN} \leq S_{l,j}^{LJ} + \eta \cdot (1 - Y_{l,j,n}) \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A, n \in \mathbf{N}_j^J \tag{3.58}$$

$$E_{l,j}^{LJ} - \eta \cdot (1 - Y_{l,j,n}) \leq E_{j,n}^{JN} \leq E_{l,j}^{LJ} + \eta \cdot (1 - Y_{l,j,n}) \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A, n \in \mathbf{N}_j^J \tag{3.59}$$

$$F_{l,j}^{LJ} - \zeta_j^U \cdot (1 - Y_{l,j,n}) \leq F_{j,n}^{JN} \leq F_{l,j}^{LJ} + \zeta_j^U \cdot (1 - Y_{l,j,n}) \quad \forall l, j \in \mathbf{J}_l \cap \mathbf{J}^A, n \in \mathbf{N}_j^J \tag{3.60}$$

Constraints (3.56)/(3.57) define the segment-customer aggregated start/end time variables $S_{l,j}^{LJ}/E_{l,j}^{LJ}$. Constraints (3.58) require that if $Y_{l,j,n} = 1$, start time $S_{j,n}^{JN}$ is equal to $S_{l,j}^{LJ}$. Similar constraints are enforced for the end time and flow amount in (3.59) and (3.60).

$$L0_j^A - \rho_j \cdot S_{j,n}^{JN} + \sum_{n' < n} F_{j,n'}^{JN} \geq \zeta_j^L \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}_j^J \tag{3.61}$$

$$L0_j^A - \rho_j \cdot E_{j,n}^{JN} + \sum_{n' \leq n} F_{j,n'}^{JN} \leq \zeta_j^U \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}_j^J \tag{3.62}$$

$$S_{j,n}^{JN} \geq E_{j,n-1}^{JN} \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}_j^J \tag{3.63}$$

Constraints (3.61) 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 (3.62) state that inventory should be lower than the upper bound after a delivery, which corresponds to one of the inventory maxima. Constraints (3.63) are the sequencing constraints for visits to a customer. These constraints in conjunction with constraints (3.50) enforce inventory bounds throughout the horizon.

3.4.8. 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 $q \in \mathbf{Q} = \{0,1,\dots,maxQ\}$, 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 Figure 3.9, 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).

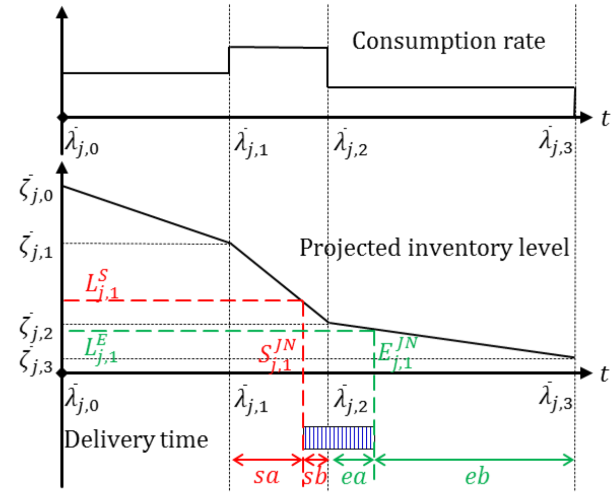
The constraints are as follows,

$$S^{JN}_{j,n} = \sum_{q \in \mathbf{Q}_j} \bar{\lambda}_{j,q} \cdot P^S_{j,n,q} \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{3.64}$$

$$L^S_{j,n} = \sum_{q \in \mathbf{Q}_j} \bar{\zeta}_{j,q} \cdot P^S_{j,n,q} \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{3.65}$$

$$L^S_{j,n} + \sum_{n' < n} F^{JN}_{j,n'} \geq \zeta^L_j \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{3.66}$$

$$\sum_{q \in \mathbf{Q}_j} P^S_{j,n,q} = 1 \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{3.67}$$



SOS2 variable values for this visit:

$P^S_{j,1,1} = \frac{sb}{sa+sb}$; $P^S_{j,1,2} = \frac{sa}{sa+sb}$; $P^S_{j,1,0} = P^S_{j,1,3} = 0$

$P^E_{j,1,2} = \frac{eb}{ea+eb}$; $P^E_{j,1,3} = \frac{ea}{ea+eb}$; $P^E_{j,1,0} = P^E_{j,1,1} = 0$

Figure 3.9. Illustration of parameters and variables introduced for piecewise linear approximation, shown by an example of the first visit to customer j.

In constraints (3.64), $P^S_{j,n,q}$ is related to $\bar{\lambda}_{j,q}$ and start time variable $S^{JN}_{j,n}$. In constraints (3.65), we calculate the projected inventory level at the start of slot n of customer j, based on $\bar{\zeta}_{j,q}$. Constraints (3.66) replace constraints (3.61) for the lower bound before a delivery. In constraints (3.67), the summation of variable $P^S_{j,n,q}$ over index q should be 1.

$$E^{JN}_{j,n} = \sum_{q \in \mathbf{Q}_j} \bar{\lambda}_{j,q} \cdot P^E_{j,n,q} \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{3.68}$$

$$L^E_{j,n} = \sum_{q \in \mathbf{Q}_j} \bar{\zeta}_{j,q} \cdot P^E_{j,n,q} \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{3.69}$$

$$L^E_{j,n} + \sum_{n' \leq n} F^{JN}_{j,n'} \leq \zeta^U_j \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{3.70}$$

$$\sum_{q \in \mathbf{Q}_j} P^E_{j,n,q} = 1 \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}^J_j \tag{3.71}$$

Constraints (3.68)-(3.71) are the counterpart of constraints (3.64)-(3.67) for the end time of a customer slot, and constraints (3.70) replace constraints (3.62) for the upper bound after a delivery.

3.4.9. Objective

Following the objective function (3.11) in the upper level VR model, we minimize the total distribution cost,

$$\min O^{SP} = \gamma^D \sum_{i,r,l \in \mathbf{L}_r^R \setminus \mathbf{L}^2} \tau_r^D \cdot X_{i,l}^{IL} + \gamma^W \sum_{i,n \in \mathbf{N}^I} \left(E_{i,n}^I - S_{i,n}^I\right) + (\gamma^R - \gamma^W \cdot \psi) \sum_{i,l \in \mathbf{L}^1} X_{i,l}^{IL} \\ + \gamma^V \sum_{i,l} |\mathbf{J}_l^L \cap \mathbf{J}^\mathbf{C}| \cdot X_{i,l}^{IL} + \gamma^X \sum_{i,l \notin \mathbf{L}^2} F_{i,l}^{SX} \tag{3.72}$$

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.

3.5. Iterative Approach

In the upper level VR subproblem (§3.3), we select the routes (and trucks to carry out the routes) to minimize cost; based on the selected routes, the lower level SP model (§3.4) 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 $LB = \max(LB, O^{VR} - \gamma^X \sum_{i,r} F_{i,r}^{RX})$; 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 $UB = \min(UB, O^{SP} - \gamma^X \sum_{i,l \notin \mathbf{L}^2} F_{i,l}^{SX})$. When LB and UB are close enough or when a predefined iteration number is reached, i.e., $(UB - LB)/LB \leq \epsilon$ 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:

No integer feasible solution can be found by SP, because (1) there are not enough drivers to carry out the routes selected in VR (since drivers are not considered in VR); or (2) some routes are not feasible for SP when scheduling constraints are considered.

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 (§3.5.1), which may lead to intractable iterations (Hooker et al., 2000; Harjunkoski et al., 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 (§3.5.3 and §3.5.4):

(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 §3.5.2). 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 Figure 3.10.

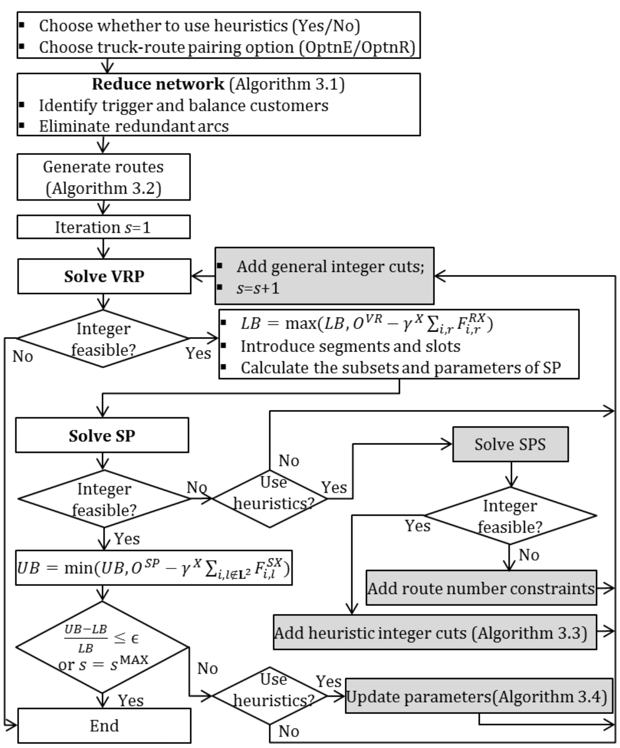


Figure 3.10. Detailed solution method flowchart; diamonds represent decision points, white boxes represent the main procedures, and grey boxes represent procedures to run before re-solving the upper level VR model. Algorithms 3.1-3.4 are shown in Appendix G.

3.5.1. 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,

$$\sum_{i,r \in \mathbf{R}^{G}_{s,i}} Z_{i,r} - \sum_{i,r \notin \mathbf{R}^{G}_{s,i}} Z_{i,r} \leq \sum_{i} \left|\mathbf{R}^{G}_{s,i}\right| - 1 \quad \forall s \tag{3.73}$$

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.

3.5.2. Truck-route Paring 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,

$$\mathbf{I}_l = \left\{ i \middle| \sum_{r \in \mathbf{R}_l} Z_{i,r} > 0 \right\} \tag{3.74}$$

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,

$$\mathbf{I}_l = \begin{cases} \left\{ i \,\middle|\, \sum_{r\in\mathbf{R}_l} Z_{i,r} > 0 \right\} & \text{if } \exists i \in \mathbf{I}\text{: } \sum_{r\in\mathbf{R}_l} Z_{i,r} = 1 \text{ and } \sum_{r\notin\mathbf{R}_l} Z_{i,r} = 0 \\ \mathbf{I} & \text{otherwise} \end{cases} \tag{3.75}$$

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.

3.5.3. 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 (3.52), (3.53), (3.61), (3.62) are replaced by constraints (3.76)-(3.79),

$$S_{i,n,k,n',l,j} + \hat{S}_{i,n,k,n',l,j} \geq \sum_m \sigma^{AS}_{j,m} \cdot W_{i,n,k,n',l,j,m} \quad \forall i, n \in \mathbf{N}^I_i, k, n' \in \mathbf{N}^K_k, l, j \in \mathbf{J}_l \cap \mathbf{J}^C \tag{3.76}$$

$$E_{i,n,k,n',l,j} - \hat{E}_{i,n,k,n',l,j} \leq \sum_{m} \sigma_{j,m}^{AE} \cdot W_{i,n,k,n',l,j,m} \;\; \forall i,n \in \mathbf{N}_i^I, k,n' \in \mathbf{N}_k^K, l,j \in \mathbf{J}_l \cap \mathbf{J}^C \tag{3.77}$$

$$L0_j^A - \rho_j \cdot S_{j,n}^{JN} + \sum_{n'<n} F_{j,n'}^{JN} + \hat{F}_{j,n}^L \geq \zeta_j^L \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}_j^J \tag{3.78}$$

$$L0_j^A - \rho_j \cdot E_{j,n}^{JN} + \sum_{n'\leq n} F_{j,n'}^{JN} - \hat{F}_{j,n}^U \leq \zeta_j^U \quad \forall j \in \mathbf{J}^A, n \in \mathbf{N}_j^J \tag{3.79}$$

The new model, which consists of constraints (3.20)-(3.51), (3.54)-(3.60), (3.63), (3.76)-(3.79), and minimizes objective function (3.72) 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 (3.80)-(3.82) 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:

$$\sum_{i,r:\tau_r^R \geq \eta/2} Z_{i,r} \leq |\mathbf{K}| \tag{3.80}$$

$$\sum_{i,r} \tau_r^W Z_{i,r} \leq |\mathbf{K}| \cdot \left\{ \left\lfloor \frac{\eta}{24} \right\rfloor \theta^W + \min(\theta^W, \eta \bmod 24) \right\} \tag{3.81}$$

$$\sum_{i,r} \tau_r^D Z_{i,r} \leq |\mathbf{K}| \cdot \left\{ \left\lfloor \frac{\eta}{24} \right\rfloor \theta^D + \min(\theta^D, \eta \bmod 24) \right\} \tag{3.82}$$

In constraint (3.80), 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 (3.81), 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 (3.82) is the counterpart of constraint (3.81) for driving time.

If SPS returns an integer feasible solution, 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 (3.16) 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 3.3 in Appendix G, and the heuristic integer cuts are generated based on the truck-route pairing option. If OptnE is adopted, we introduce infeasible truck set $\mathbf{I}_s^E$ and infeasible route set $\mathbf{R}_{i,s}^E$ and add the following constraints,

$$\sum_{r \in \mathbf{R}_{i,s}^E} Z_{i,r} \leq \left|\mathbf{R}_{i,s}^E\right| - 1 \quad \forall s, i \in \mathbf{I}_s^E \tag{3.83}$$

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

$$\sum_{i,r \in \mathbf{R}_s^R} Z_{i,r} \leq |\mathbf{R}_s^R| - 1 \quad \forall s \tag{3.84}$$

to exclude the infeasible route combinations for all trucks.

3.5.4. 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 $\tau x_{i,r}$ from the SP solution, objective function (3.11) and constraints (3.15) of the original VR model are modified as follows,

$$\min \sum_{i,r} [(\gamma_r^R + \gamma^W \cdot \tau x_{i,r}) Z_{i,r} + \gamma^X F_{i,r}^{RX}] \tag{3.85}$$

$$\sum_r (\tau_r^R + \tau x_{i,r}) Z_{i,r} \le \eta, \quad \forall i \tag{3.86}$$

Algorithm 3.4 in Appendix G summarizes the parameter updating procedure.

3.6. 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 hours, check-in/out time is 0.5 hours, loading/delivering time at the plant/customers is 1 hour, minimum resting time is 10 hours, and maximum daily driving/working time is 11/14 hours. 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-hour 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 hours increases significantly. We tested all the problems using 4 different options (combinations of truck-route parings and heuristics), as summarized in Table 3.1.

Table 3.1. 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

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 seconds for each mathematical program. The termination criterion, ϵ, was 0.005. Also, the iterative procedure was terminated after 20 iterations.

3.6.1. 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 Figure 3.11, the data for customers and trucks are given in Tables 3.2-3.3, and iterations and solution time are summarized in Table 3.4.

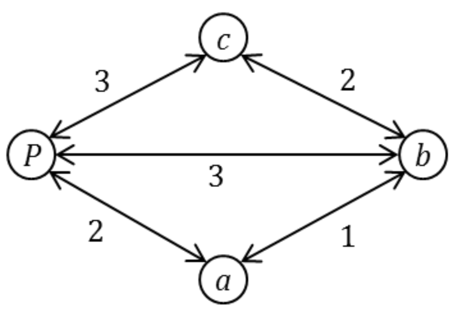


Figure 3.11. Network structure for the toy example, *P* is the plant, *a, b, c* are three customers.

Table 3.2. 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 3.3. Truck capacities for the toy example.

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

Table 3.4. 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*->*c*->*P,* arrives at customer *c* at time 6, and delivers 570 units of product; truck T2 takes route *P*->*b*->*a*->*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 minutes to solve this toy problem and prove optimality using a full IRP model (shown in Chapter 2), 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*->*a*->*b*->*c*->*P* or *P*->*c*->*b*->*a*->*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*->*a*->*b*->*c*->*P* or *P*->*c*->*b*->*a*->*P*, and the LB/UB are $454/+\infty$. 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 3.4, 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*->*a*->*b*->*c*->*P* and *P*->*c*->*b*->*a*->*P*, when option 4 is used. In iteration 3, routes *P*->*c*->*P* and *P*->*b*->*a*->*P* are selected, and the iterative procedure ends ($LB = O^{VR} = 662$, $O^{VR} = 665$, $UB = 662$).

3.6.2. 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 (§3.2), there are typically fewer than 35 customers (including 0-2 order-only customers). 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 3.5 shows the overall algorithm performance, including instance sizes, iteration numbers, total solution time and objective values. Model statistics are shown in Tables 3.6-3.8, 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 (shown in Chapter 2). The corresponding solution statistics for the smaller instances are given in Table 3.9.

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 hours, 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 minutes 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.

Table 3.5. Instance characteristics, iterations, solution times, and objective function values using options 1-4.

Ins-tance	Custo-mers	Tru-cks	Dri-vers	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 3.6. 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	16028	3376	21578	77664	1	3.17
8	19712	4130	25993	94373	480	7.57
9	683	171	796	2875	1	0.33
10	22764	5054	26491	94584	1528	11.67
11	60470	12810	69828	258774	936	61.03
12	7224	1659	8511	29867	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 3.7. 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	10836	1	0.39
5	3339	738	4753	17527	1	2.70
6	18879	3287	25233	104764	1275	286.11
7	3154	830	4614	16792	1	2.82
8	6112	1406	8820	32480	2762	153.36
9	2389	589	3303	12740	1	3.23
10	9392	1591	13231	50539	932	78.25
11	8201	1540	11285	44042	1	22.37
12	7076	1345	9927	37358	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 3.8. 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	11855	45404	1	4.31
5	13528	3022	18962	71200	1	20.76
6	60850	9508	82717	337597	19	300.73
7	12768	3514	18226	69062	1	14.52
8	41608	9864	58022	220007	88	300.52
9	14103	3368	19480	78145	1	20.12
10	94907	11020	94907	368356	1	300.15
11	14103	3368	19480	78145	1	18.88
12	50442	9727	69732	266266	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 3.9. 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 hours	Objective value after 20 hours	Gap after 20 hours
1	16328	10445	8511	128029	342	774.0	104889	666.0	18%
2	30852	18260	15576	279301	430	1459.3	39005	924.0	39%
3	10130	6838	5443	81636	52	1922.6	237401	1496.4	11%
4	43222	25132	21959	406516	620	1734.2	22157	1186.0	60%

For smaller problems (instances 1-4), the algorithm is finished within 10 seconds 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 seconds for instance 6 using option 3. To further study this, we tested all instances with a 1200-second time limit for solving SP; OptnE still outperformed OptnR.

(b) Option 2 is the fastest; all instances were solved within 7 minutes. 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%).

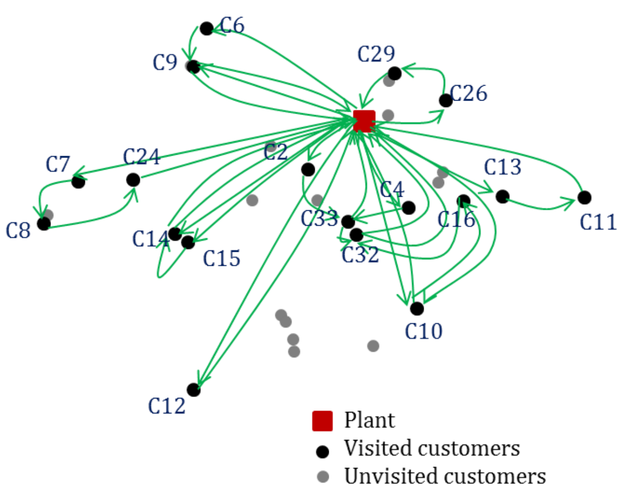


Figure 3.12. Routes selected for instance 11.

Finally, we show the routing and scheduling solution of instance 11, which was also used as the example in §3.2.3. Figure 3.12 shows the routing decisions (note that some balance customers are not visited in the planning horizon) and Figure 3.13 shows the final solution as a Gantt chart.

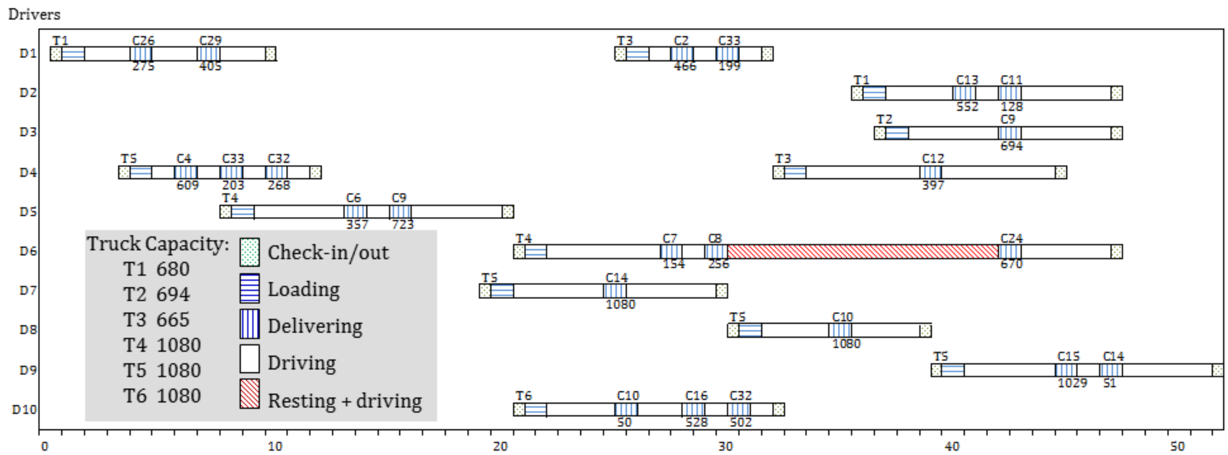


Figure 3.13. Gantt chart showing the solution for instance 11.

3.6.3. 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 minutes, and the time increases as the numbers of trucks, arcs and routes increase (Tables 3.5 and 3.6). 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 Figure 3.14, 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 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: (1) The solution time limit should be a function of the numbers of trucks, customers, drivers, and selected routes in VR. (2) When SP is not solved within the time limit, we do not use the heuristics shown in Figure 3.10, and increase the time limit for the following iterations. (3) 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.

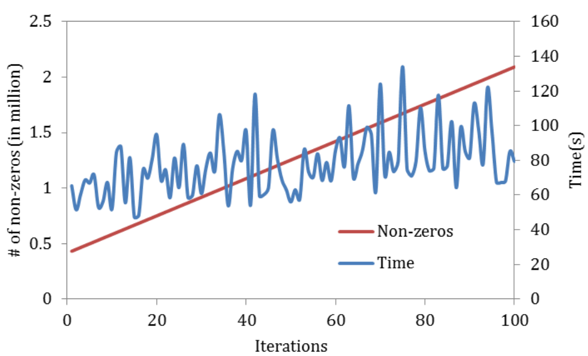


Figure 3.14. Effects of integer cuts on the number of non-zeros and solution time.

3.7. Conclusions

In this chapeter, 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.

3.8. 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
$\mathbf{A}_{r,j}^{RP}$	arcs traveled before j in route r
$\mathbf{I}_l$	trucks that can carry out segment l

$\mathbf{I}_s^E$	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*
$\mathbf{J}_l^{start}/\mathbf{J}_l^{end}$	start/end SC node of segment *l*
$\mathbf{J}_r$	customers visited in route *r*
$\mathbf{J}^T/\mathbf{J}^B$	trigger/balance customers
$\mathbf{J}_j^R$	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*
$\mathbf{L}_l^{next}$	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*
$\mathbf{R}_{s,i}^G/\mathbf{R}_{s,i}^E/\mathbf{R}_s^R$	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
$\zeta_j^L/\zeta_j^U/\zeta_j^S$	minimum/ maximum/ safety level of anticipatable customer *j*

$\bar{\zeta}_{j,q}$	projected inventory level at point q of customer *j* without deliveries
η	planning horizon
θ^W/θ^D	maximum daily working/driving time
$\bar{\lambda}_{j,q}$	time at point q of customer *j*
ξ_i	capacity of truck *i*
ρ_j	constant consumption rate of anticipatable customer *j*
$\rho_j^T(t)$	consumption rate of anticipatable customer *j* at time t
$\sigma_{j,m}^{AS}/\sigma_{j,m}^{AE}$	start/end time of access window m of customer *j*
$\sigma_j^{OS}/\sigma_j^{OE}$	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*
$L0_j^A$	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
$\sigma_j^{\text{MIN}}/\sigma_j^{\text{MAX}}$	minimum/maximum demand in the planning horizon of customer *j*
$\tau_r^W/\tau_r^D/\tau_r^R$	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

$F_{i,r}^{RX}$	unused capacity of truck *i* when carrying out route *r*
$F_{i,r,j}^{R}$	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*
$X_{i,n}^{I}$	=1 if and only if slot *n* of truck *i* is used
$X_{k,n}^{K}$	=1 if and only if slot *n* of driver *k* is used
$X_{i,l}^{IL}$	=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

$P_{j,n,q}^{S}/P_{j,n,q}^{E}$	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*
$F_{l,j}^{LJ}$	delivery amount on segment *l* to customer *j*
$F_{j,n}^{JN}$	delivery amount to customer *j* at slot *n*
$\hat{F}_{j,n}^{L}/\hat{F}_{j,n}^{U}$	inventory lower/upper bound violation for customer *j* at slot *n*
$F_{i,l}^{SX}$	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*

Chapter 4

Policy Analysis based on Reoptimization for MIRP under Uncertainty[4]

In reality, IRP is a "dynamic" problem in which new information (e.g. newly forecasted production/consumption rate) arrives continuously and disruptive events (e.g., delays due to bad weather) are common. However, most models and solution methods have been developed for the static IRP/MIRP. Accordingly, in this chapter, we propose a reoptimization framework that allows us to: (1) study the impact of different sources of uncertainty on the closed-loop (implemented) solution, as opposed to the open-loop solution; and (2) study how different policies impact the quality of the closed-loop solution. In the MIRP we studied, the vessels may have specific capacities and can be rented in different modes; for all the consumption nodes and some of the production nodes, the inventory levels are monitored and should be maintained within specified bounds; for other production nodes, orders should be picked up within pre-defined time windows.

The chapter is structured as follows. In Section 4.1, we discuss background on MIRP and reoptimization. In Section 4.2, we propose a MIP model based on a discrete-time representation. In Section 4.3, we discuss the sources of uncertainty we consider and the stochastic simulation we employ to study them. In Section 4.4, we present the reoptimization framework, and in Section 4.5 we describe the different policies that can be adopted. Case studies are presented in Section 4.6. We use uppercase bold Latin letters for sets, lowercase italic Latin letters for indices, lowercase italic Greek letters for general parameters and uppercase italic Latin letters for optimization variables. Subsets are represented by the letter of the superset and a superscript. Finally, parameters which represent the history of the system, and consequently determine the state of the system at the current time, are denoted by uppercase italic Latin letters with a hat.

---

[4] This chapter is modified from Dong et al., submitted.

4.1. Background

4.1.1. Distribution Supply Chain

In MIRP, there are generally two subsets of nodes (ports): production and consumption nodes. Materials are distributed among these nodes by sea-going vessels. Traveling from one node to another is referred to as a trip. For most nodes, inventory levels and estimates of future consumption and production profiles are known. Additionally, to study features that are often found in practice, we consider owned and third-party nodes (i.e., nodes not owned by the decision-maker) as well as long- and short-term renting modes.

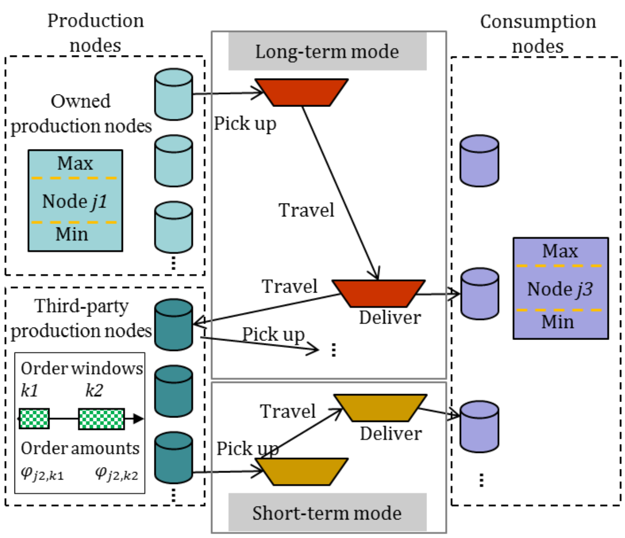


Figure 4.1. An illustration of the overall distribution network.

For all consumption nodes and the owned production nodes, the estimated future consumption and production profiles are known, and the inventory levels should be kept within their upper and lower bounds by delivering or picking up materials. For third-party production nodes, orders with specified pick-up windows are given. Material distribution can be carried out by vessels either rented in long- or short-term mode. The distribution problem is illustrated in Figure 4.1. Sources of uncertainty include vessel availability, trip delays, pick-up window specifications, consumption/production rate variations, etc.

Most pick-ups and deliveries are completed by renting a seagoing vessel and its crew *in long-term mode*. Once rented in long-term mode, a vessel needs to be held for at least a certain number of days (normally 30). Vessel availability needs to be checked with the marine vessel company before it is rented: the earlier an inquiry (call) is made, the higher the probability that a vessel will be available and successfully reserved. The advantages of long-term rental are that (1) a vessel is guaranteed to be available once rented, and (2) the decision maker can change its routes and schedules. The disadvantage is that a fee has to be paid for as long as it is rented, regardless of its utilization level.

A *short-term* rental is for a single node-to-node trip (e.g., pick up 1000 kilotons of material A at production node P1 on October 15th and deliver it to consumption node C1 on October 25th) and the rate charged is per trip. The decision maker also has to call the vessel company to make the reservation. The advantage of short-term mode is that the decision maker only pays for the vessel when it is in use. The disadvantages are: (1) a vessel may not be immediately available; (2) the per-period cost is higher compared to that for the long-term mode; and (3) a scheduled trip cannot be changed to react, for example, to a new order or trip delay.

4.1.2. Reoptimization

Reoptimization, or reactive optimization, is needed when the horizon is finite and/or the system is subject to uncertainty. If reoptimization focuses only on revising the decisions that are already made, it results in a *shrinking-horizon* approach; while if the horizon is rolled forward so that additional decisions are made, it results in a *rolling-horizon* approach. In the field of production scheduling (Méndez et al., 2006; Li and Ierapetritou, 2008; Verderame et al., 2010; Harjunkoski et al., 2014), reoptimization has been studied (Vieira et al., 2003; Ouelhadj and Petrovic, 2009; Gupta et al., 2016). Upon obtaining new information or observing a disturbance, (part of) the unexecuted schedule is recomputed. Rescheduling approaches can be broadly divided into deterministic

(Bassett et al., 1997; Novas and Henning, 2010; Gupta and Maravelias, 2016) and stochastic (Balasubramanian and Grossmann, 2004; Janak et al., 2007; Cui and Engell, 2010). Deterministic approaches are easier to implement and computationally less demanding, while stochastic approaches can potentially lead to better solutions. A state space model explicitly accounting for disturbances for chemical production rescheduling has been proposed (Subramanian et al., 2012). An optimization framework of rolling horizons has been proposed for the vendor managed inventory systems (Al-Ameri et al., 2008), while make-to-order polices have been analyzed on a rolling horizon basis for order-based production SCs (Sahin et al., 2008).

In this chapter, we employ a deterministic approach and adopt a rolling horizon framework for reoptimization. Each solution obtained from solving the MIP model is referred to as an *open-loop* solution. The first *move* of this solution, that is, the decisions corresponding to the first period (t = 1) are implemented. After the horizon is rolled forward, uncertainty is observed and feedback is incorporated, and the model is re-solved; the decisions for t = 1 are implemented and the process is repeated. The final, implemented solution is referred to as the *closed-loop* solution.

4.1.3. Problem Statement

We consider a sufficiently large fleet of vessels that can be rented in long-term mode. Unrented vessels are located in a pseudo node, the *vessel center* (*vc*). We do not model vessels rented in short-term mode explicitly because it is unnecessary to keep track of their location (since they are rented for single trips); we simply model the time and the pick-up quantity.

We consider the following indices and sets:

(a) $i \in \mathbf{I}$: shipping vessels (that can be rented in long-term mode).

(b) $j \in \mathbf{J}^P$: production nodes, which can be classified as owned, $j \in \mathbf{J}^{OP}$, and third-party, $j \in \mathbf{J}^{TP}$.

(c) $j \in \mathbf{J}^C$: consumption nodes.

(d) $j \in \mathbf{J} = \mathbf{J}^P \cup \mathbf{J}^C \cup \{vc\}$: all nodes in the supply chain network.

(e) $k \in \mathbf{K}_j$: orders from node $j \in \mathbf{J}^{TP}$.

(f) $m \in \mathbf{M}$: materials.

(g) $\mathbf{A} \subseteq \mathbf{J} \times \mathbf{J}$: arcs in the SC network.

The following data are assumed to be available:

(a) γ_i^{MAX}: capacity of vessel *i*.

(b) γ_i^{MIN}: lower bound on the load of vessel *i* when traveling from a production node to a consumption node.

(c) $\bar{\tau}_{jj'}$: traversal time along arc (*j,j'*), which includes the travel time, plus the pick-up/delivery time at node *j*.

(d) $\xi_{jj'}^{MAX}$: maximum load along arc (*j,j'*), which is determined by the waterline bound along the arc.

(e) $\zeta_{jm}^{MAX}/\zeta_{jm}^{MIN}$: maximum/minimum inventory level for material *m* in consumption node or owned production node *j*; inventory violations leading to overflow or underflow are allowed but penalized.

(f) φ_{jmk}: amount of material *m* in order *k* from node $j \in \mathbf{J}^{TP}$.

(g) $\sigma_{jk}^{OS}/\sigma_{jk}^{OE}$: start/end time of pick-up windows for order *k* from node $j \in \mathbf{J}^{TP}$.

(h) The forecast production or consumption profile for node $j \in \mathbf{J}^{OP} \cup \mathbf{J}^{C}$ (discussed in §4.2).

We are also given the initial state of the system (see §4.2.2 for details), including node inventories, vessel initial loads, and vessel location information.

The objective is to minimize the total distribution cost, which includes material holding, overflow, underflow, vessel renting, and transportation cost. The corresponding parameters are as follows:

(a) ϑ^{L}: minimum number of periods for which each vessel can be rented in long-term mode.

(b) π_i^{FL}: fixed cost for renting vessel *i* for the minimum number of periods.

(c) π_i^{DL}: additional cost for extending the long-term rental of vessel *i*.

(d) $\pi_{jj'}^{S}$: cost of renting a vessel in short-term mode to serve (*j,j'*).

(e) $\pi_{jj'}^{FT}$: fixed transportation cost for trip along (*j, j'*).

(f) $\pi_{jj'}^{VT}$: unit variable transportation cost for trip (*j, j'*).

(g) π_{jm}^{MH}: unit holding cost for material *m* at node *j* for one time period.

Note that parameters (b)-(d) include the cost of renting the vessels as well as their crew members.

4.2. Discrete-time Model

We adopt a discrete-time approach, where the planning horizon η is divided into periods with uniform length δ. Time points and time periods are both denoted by $t \in \mathbf{T}$. Period t starts at point *t*-1 and ends at point *t*, so the planning horizon has η/δ periods (1, 2, …, η/δ), and $\eta/\delta + 1$ points (0, 1, …, η/δ). Vessels start trips at time points, while they travel along arcs or stay at nodes during time periods. The consumption/production rate of each node during each time period is known, and its inventory level is modeled; it represents the level just after the pick-up/delivery occurring at the same time point.

The traversal time along an arc is rounded up, $\tau_{jj'} = \lceil \overline{\tau}_{jj'}/\delta \rceil$ to ensure that the solution is feasible. The approximation can be modified, if it is desirable to be less conservative. For each node *j*, we use ρ_{jmt} to denote the production or consumption rate of material *m* during period *t*, which is positive for production nodes ($j \in \mathbf{J}^P$), and negative for consumption nodes ($j \in \mathbf{J}^C$). For a third-party production node $j \in \mathbf{J}^{TP}$, orders $k \in \mathbf{K}_j$ are placed (described by parameters φ_{jmk} and $\sigma_{jk}^{OS}/\sigma_{jk}^{OE}$). To treat nodes uniformly, if *j* is a third-party production node, the production rate in the period just before the start time of a pick-up window is equal to the order amount,

$$\rho_{jmt} = \begin{cases} \varphi_{jmk} & \text{if } \exists k \in \mathbf{K}_j : t = \lceil \sigma_{jk}^{OS}/\delta \rceil \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

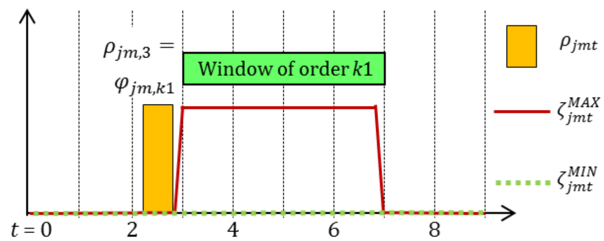


Figure 4.2. Production rate, upper and lower bounds on inventory levels for a third-party production node.

We use $\zeta_{jmt}^{MIN}/\zeta_{jmt}^{MAX}$ to denote the lower/upper bound on the inventory level of material m in node j at time t. For consumption or owned production nodes, $\zeta_{jm}^{MIN}/\zeta_{jm}^{MAX}$ represent the physical constraints (see §2.3). For third-party production nodes, ζ_{jmt}^{MIN} is always 0; while ζ_{jmt}^{MAX} is the order amount if $\lceil \sigma_{jk}^{OS}/\delta \rceil \leq t < \lfloor \sigma_{jk}^{OE}/\delta \rfloor$ for an order k, and 0 otherwise (see Figure 4.2).

$$\zeta_{jmt}^{MIN} = \begin{cases} \zeta_{jm}^{MIN} & \text{if } j \in \mathbf{J}^{OP} \cup \mathbf{J}^{C} \\ 0 & \text{if } j \in \mathbf{J}^{TP} \end{cases} \tag{4.2}$$

$$\zeta_{jmt}^{MAX} = \begin{cases} \zeta_{jm}^{MAX} & \text{if } j \in \mathbf{J}^{OP} \cup \mathbf{J}^{C} \\ \varphi_{jmk} & \text{if } j \in \mathbf{J}^{TP} \text{ and } \exists k \in \mathbf{K}_j : \lceil \sigma_{jk}^{OS}/\delta \rceil \leq t < \lfloor \sigma_{jk}^{OE}/\delta \rfloor \\ 0 & \text{if } j \in \mathbf{J}^{TP} \text{ and } \nexists k \in \mathbf{K}_j : \lceil \sigma_{jk}^{OS}/\delta \rceil \leq t < \lfloor \sigma_{jk}^{OE}/\delta \rfloor \end{cases} \tag{4.3}$$

4.2.1. Variables

An overhead bar is used for variables representing quantities or states *during a time period t*, while no bar is placed for variables representing quantities or actions *at a time point*. We define the following binary variables:

(a) $W_{ijj't}^{L} = 1$ if vessel i starts a trip from j to j' at time point t.

(b) $\bar{X}_{ijt}^{L} = 1$ if vessel i is at node j during time period t; when not rented, $\bar{X}_{i,vc,t}^{L} = 1$.

(c) $W_{jj't}^{S} = 1$ if a vessel in short-term mode starts a trip from j to j' at time point t.

and the following nonnegative continuous variables to represent material flows and inventory levels:

(a) $F_{ijj'mt}^{L}$: amount of material m loaded in vessel i that starts to travel from j to j' at t.

(b) $F^{S}_{jj'mt}$: amount of material *m* loaded in short-term rental, from *j* to *j'* starting at *t*.

(c) L_{jmt}: inventory level of material *m* in node *j* at time point *t*.

(d) $L^{OF}_{jmt}/L^{UF}_{jmt}$: amount over/below the upper/lower bound on the inventory level of material *m* in node *j* at time point *t*.

Figure 4.3 illustrates the aforementioned variables based on a delivery to a consumption node by a vessel in long-term mode.

Finally, we use the following variables to model different types of cost:

(a) $\bar{Y}^{L}_{it} = 1$ if vessel *i* is used during period *t* beyond the minimum ϑ^{L} periods.

(b) C^{MH}_{t}: material holding cost.

(c) C^{OF}_{t}/C^{UF}_{t}: overflow/underflow cost.

(d) C^{FT}_{t}: fixed transportation cost.

(e) C^{VT}_{t}: variable transportation cost.

(f) C^{FL}_{t}: renting cost for the minimum ϑ^{L}-period in long-term mode.

(g) C^{EL}_{t}: renting cost for extended time in long-term mode.

(h) C^{S}_{t} : short-term renting cost.

(i) C^{ALL}: total cost.

4.2.2. System Initial States

An overhead bar is used for variables representing quantities or states *during a time period t*, while no bar is placed for variables representing quantities or actions *at a time point*. We define the following binary variables:

We define the system initial state (t = 0) using the following:

(a) $\hat{L}_{jm0}$: initial inventory of node *j*.

(b) $\hat{X}^L_{ijt}$: initial location of vessel i; if located at j initially, then $\hat{X}^L_{ij0} = 1$; otherwise, $\hat{X}^L_{ij0} = 0$; if traveling initially and will arrive at j at t, then $\hat{X}^L_{ijt} = 1$; otherwise, $\hat{X}^L_{ijt} = 0$.

(c) $\hat{F}^L_{ijmt}/\hat{F}^S_{j'jmt}$: amount of en route material m, arriving at consumption node j at time point t, from vessel i or from production node j' using a short-term rental; the source of this incoming flow is monitored in case it needs to be modified due to trip delay.

(d) $\widehat{W}^L_{ijj't}/\widehat{W}^S_{jj't}$: previously made travel decisions (which should be respected in the solution); $\widehat{W}^L_{ijj't} = 1$means that vessel i should leave j for j' at time t, while $\widehat{W}^S_{jj't} = 1$ means that a vessel in short-term mode should leave j for j' at time t.

The letters used in these parameters are the same as the optimization variables (e.g., $\hat{L}_{jm0}$ represents initial inventory level and L_{jmt} is an inventory level variable), though sometimes the parameters have slightly different meaning (e.g., $\hat{F}^L_{ijmt}$ is the amount of material arriving at node j in vessel i at time point t, while $F^L_{ijj'mt}$ is the amount of material loaded in the vessel, if it starts to travel from j to j' at t).

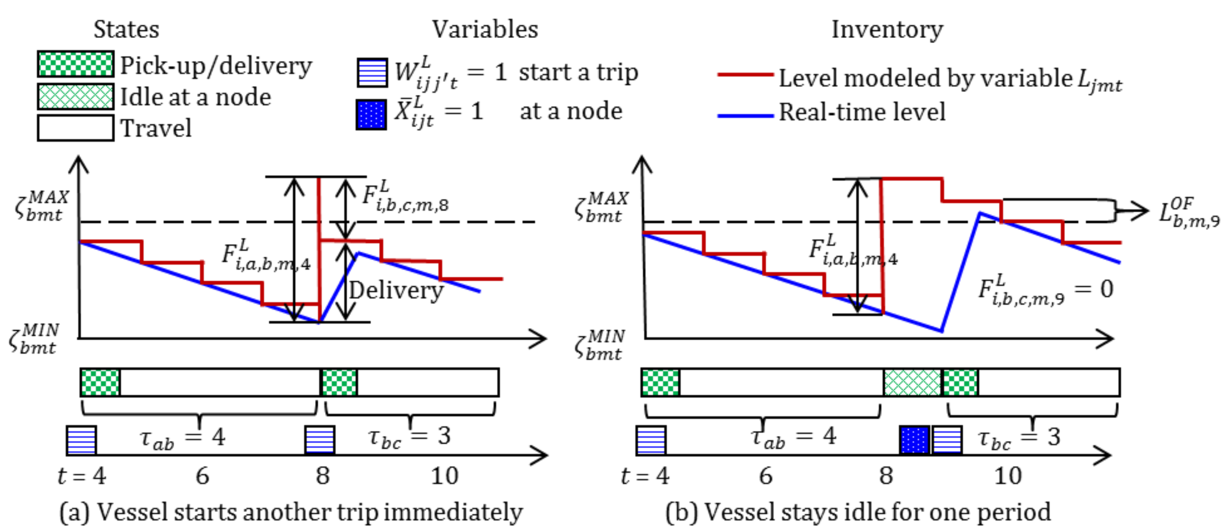


Figure 4.3. Inventory modeling for consumption node b, with a vessel traveling on $a \rightarrow b \rightarrow c$. The delivery time is included at the beginning of each traversal time. (a) The vessel arrives at b, immediately makes delivery, and leaves. (b) The vessel stays at b for one period before making delivery.

4.2.3. Constraints

Vessel location. The following equation models the vessel location (including departing from and returning to the vessel center):

$$\bar{X}^L_{ijt} = \bar{X}^L_{ij,t-1} + \sum_{j'} W^L_{ij'j,t-\tau_{j'j}-1} - \sum_{j'} W^L_{ijj',t-1} + \hat{X}^L_{ij,t-1} \quad \forall i,j,t \tag{4.4}$$

Vessel i is at node j in period t, if (1) it was there in the previous period ($\bar{X}^L_{ij,t-1} = 1$) or it has just arrived (either $\sum_{j'} W^L_{ij'j,t-\tau_{j'j}-1} = 1$ or $\hat{X}^L_{ij,t-1} = 1$), and (2) it does not leave at the start of period t. Note that $\bar{X}^L_{i,vc,t}$=1 represents that vessel i is not rented during t (i.e., located at the vessel center).

Arc flow. Constraints (4.5) enforce that the vessel capacities and maximum load along arcs are respected, and constraints (4.6) enforce that to avoid routes with small pick-ups, the flow amount from a production node to a consumption node should be greater than a minimum amount.

$$\sum_m F^L_{ijj'mt} \leq \min\left(\xi^{MAX}_{jj'}, \gamma^{MAX}_i\right) W^L_{ijj't} \quad \forall i,(j,j') \in \mathbf{A}, t \tag{4.5}$$

$$\sum_m F^L_{ijj'mt} \geq \gamma^{MIN}_i W^L_{ijj't} \quad \forall i, j \in \mathbf{J}^P, j' \in \mathbf{J}^C, (j,j') \in \mathbf{A}, t \tag{4.6}$$

Similarly, in short-term mode, the arc flow is subject to constraints (4.7) and (4.8).

$$\sum_m F^S_{jj'mt} \leq \min(\xi^{MAX}_{jj'}, \gamma^{MAX}) W^S_{jj't} \quad \forall (j,j') \in \mathbf{A}, t \tag{4.7}$$

$$\sum_m F^S_{jj'mt} \geq \gamma^{MIN} W^S_{jj't} \quad \forall j \in \mathbf{J}^P, j' \in \mathbf{J}^C, (j,j') \in \mathbf{A}, t \tag{4.8}$$

For simplicity, the maximum/minimum pick-up amounts of vessels in short-term mode, γ^{MAX}/ γ^{MIN}, are assumed to be the same for all vessels, but this assumption can be relaxed.

Inventory level. The inventory level L_{jmt} is equal to the inventory level at the previous time point plus production ($\rho_{jmt} > 0$) or consumption ($\rho_{jmt} < 0$) plus incoming flows minus outgoing flows,

$$L_{jmt} = L_{jm,t-1} + \rho_{jmt} + \sum_{i,j'} F^{L}_{ij'jm,t-\tau_{j'j}} + \sum_{j'} F^{S}_{j'jm,t-\tau_{j'j}} - \sum_{i,j'} F^{L}_{ijj'mt} - \sum_{j'} F^{S}_{jj'mt} + \sum_{i} \hat{F}^{L}_{ijmt} + \sum_{j'} \hat{F}^{S}_{j'jmt} \quad \forall j \in \mathbf{J}^P \cup \mathbf{J}^C, m, t \tag{4.9}$$

where the last two terms represent the incoming flow to *j* that are initially en route. Inventory levels are constrained as follows,

$$L_{jmt} - L^{OF}_{jmt} \leq \zeta^{MAX}_{jmt} + \sum_{i} \gamma^{MAX}_{i} \bar{X}_{ijm,t+1} \quad \forall j \in \mathbf{J}^P \cup \mathbf{J}^C, m, t \tag{4.10}$$

$$L_{jmt} + L^{UF}_{jmt} \geq \zeta^{MIN}_{jmt} \quad \forall j \in \mathbf{J}^P \cup \mathbf{J}^C, m, t \tag{4.11}$$

where overflow/underflow amounts are denoted by $L^{OF}_{jmt}/L^{UF}_{jmt}$. The second term on the right hand side (RHS) of constraints (4.10) is needed because when a vessel is at a node, the amount of material in the vessel is included in variable L_{jmt} (see Figure 4.3(b)).

4.2.4. Objective Function

The objective is to minimize the overall distribution cost,

$$\min C^{ALL} = \sum_{t} (C^{MH}_{t} + C^{OF}_{t} + C^{UF}_{t} + C^{FT}_{t} + C^{VT}_{t} + +C^{FL}_{t} + C^{EL}_{t} + C^{S}_{t}) \tag{4.12}$$

The material holding cost is calculated as follows,

$$C^{MH}_{t} = \sum_{j,m} \pi^{MH}_{jm} L_{jmt} \quad \forall t \tag{4.13}$$

where π^{MH}_{jm} is the unit holding cost for material *m* in node *j*.

The overflow and underflow cost are calculated as follows,

$$C^{OF}_{t} = \sum_{j,m} \pi^{OF}_{jmt} L^{OF}_{jmt} \quad \forall t \tag{4.14}$$

$$C^{UF}_{t} = \sum_{j,m} \pi^{UF}_{jmt} L^{UF}_{jmt} \quad \forall t \tag{4.15}$$

where π_{jmt}^{OF} and π_{jmt}^{UF} are the unit inventory violation cost in each period for material *m* in node *j*.

Transportation cost includes fixed cost, which is independent of the load, and variable cost:

$$C_t^{FT} = \sum_{i,j,j'} \pi_{jj'}^{FT} W_{ijj't}^{L} + \sum_{j,j'} \pi_{jj'}^{FT} W_{jj't}^{S} \quad \forall t \tag{4.16}$$

$$C_t^{VT} = \sum_{i,j,j',m} \pi_{jj'}^{VT} F_{ijj'mt}^{L} + \sum_{j,j',m} \pi_{jj'}^{VT} F_{jj'mt}^{S} \quad \forall t \tag{4.17}$$

where $\pi_{jj'}^{FT}$ is the fixed cost for traveling along arc (*j,j'*), and $\pi_{jj'}^{VT}$ is the unit variable cost.

Long-term renting has two cost components: one for the renting over the minimum ϑ^L periods, and another for extending the renting beyond ϑ^L periods. The binary variable $\bar{Y}_{it}^{L}$, representing whether vessel *i* is rented beyond the minimum periods in time *t*, is constrained as follows,

$$\bar{Y}_{it}^{L} \geq \sum_{j \in \mathbf{J}^P \cup \mathbf{J}^C} \bar{X}_{ijt}^{L} + \sum_{j,j',t-\tau_{jj'} \leq t' \leq t-1} W_{ijj't'}^{L} - \sum_{j,t-\vartheta^L \leq t' \leq t-1} W_{i,vc,jt'}^{L} - \chi_{it} \quad \forall i,t \tag{4.18}$$

where parameter $\chi_{it} = 1$ if (1) vessel *i* is already rented at the start of the horizon, and (2) period *t* is within the first ϑ^L periods since this renting. Therefore, vessel *i* is rented beyond the minimum number of periods, if it is still rented (one of the first two summation is 1) and the current rental started more than ϑ^L periods ago (last two terms are both 0). This constraint is illustrated in Figure 4.4.

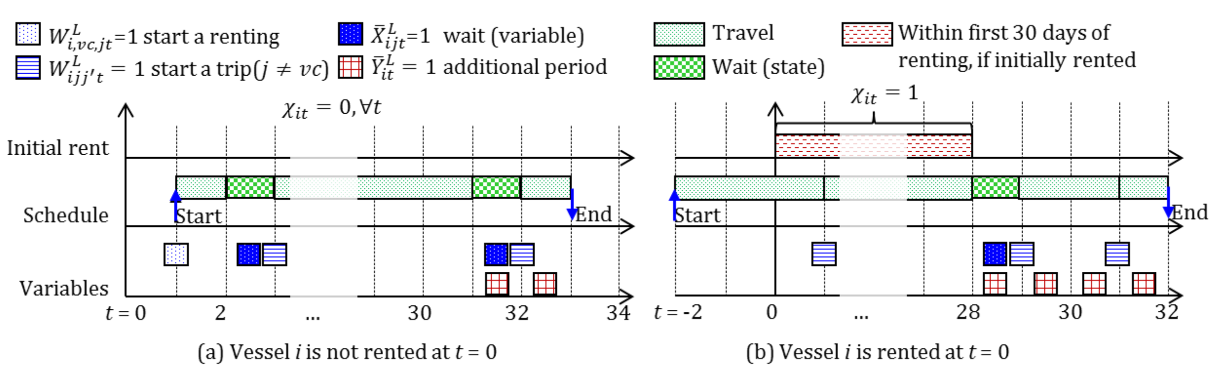


Figure 4.4. Modeling for the extended renting of vessels beyond ϑ^L periods; $\vartheta^L = 30$.

With $\bar{Y}_{it}^{L}$ defined above, the two components of cost are expressed in (4.19) and (4.20) respectively,

$$C_t^{FL} = \sum_{i,j} \pi_i^{FL} W_{i,vc,jt}^{L} \quad \forall t \tag{4.19}$$

$$C_t^{EL} = \sum_{i} \pi_i^{EL} \bar{Y}_{it}^{L} \quad \forall t \tag{4.20}$$

where π_i^{FL} is the fixed renting cost, and π_i^{EL} is the extended renting cost for one period.

The short-term renting cost is calculated as follows,

$$C_t^{S} = \sum_{j,j'} \pi_{jj'}^{S} \cdot W_{jj't}^{S} \quad \forall t \tag{4.21}$$

where $\pi_{jj'}^{S}$ is the cost for renting a vessel in short term mode to serve trip (*j,j'*), and $\pi_{jj'}^{S}$ normally depends on the trip length $\tau_{jj'}$ linearly.

4.2.5. Valid Inequalities

When no inventory violations are allowed, we can add valid inequalities to tighten the model. For third-party production node $j \in \mathbf{J}^{TP}$ with orders $\mathbf{K}_j$, binary parameter θ_{jkt}, which represents whether a time period is within a pick-up window, is defined as follows,

$$\theta_{jkt} = \begin{cases} 1 & \text{if } \lceil \sigma_{jk}^{OS}/\delta \rceil + 1 \le t \le \lfloor \sigma_{jk}^{OE}/\delta \rfloor \\ 0 & \text{otherwise} \end{cases} \tag{4.22}$$

There should be at least one outgoing arc during the pick-up window,

$$\sum_{i,j',t:\theta_{jkt}=1} W_{ijj't}^{L} + \sum_{j',t:\theta_{jkt}=1} W_{jj't}^{S} \ge 1 \quad \forall j \in \mathbf{J}^{TP}, k \in \mathbf{K}_j \tag{4.23}$$

For owned production node $j \in \mathbf{J}^{OP}$, the minimum number of pick-ups till time t should satisfy,

$$\sum_{i,j',t'\le t} W_{ijj't'}^{L} + \sum_{j',t'\le t} W_{jj't'}^{S} \ge \left\lceil \frac{\sum_m \left(\hat{L}_{jm0} + \sum_{t'\le t} \rho_{jmt'} - \zeta_{jmt}^{MAX}\right)}{\max\{\max_i \gamma_i^{MAX}, \gamma^{MAX}\}} \right\rceil \quad \forall j \in \mathbf{J}^{OP}, t \tag{4.24}$$

where the RHS represents the minimum number of pick-ups (so that the maximum inventory level is respected). The numerator is the initial inventory plus the production amount till time t minus the upper bound on inventory level, while the denominator is the largest vessel capacity.

Following the same logic, we can write constraints (4.25) for consumption nodes,

$$\sum_{i,j',t'\leq t-\tau_{j'j}} W^{L}_{ijj't'} + \sum_{j',t'\leq t-\tau_{j'j}} W^{S}_{jj't'} \geq \left\lceil \frac{\sum_m \left(\zeta^{MIN}_{jmt} - \hat{L}_{jm0} - \sum_{t'\leq t} \rho_{jmt'} - \sum_{i,t'\leq t} \hat{F}^{L}_{ijmt'} - \sum_{j',t'\leq t} \hat{F}^{S}_{j'jmt'}\right)}{\max\{\max_i \gamma^{MAX}_i , \gamma^{MAX}\}} \right\rceil \quad \forall j \in \mathbf{J}^C, t \tag{4.25}$$

where the numerator on the RHS is the minimum delivery amount to meet the lower bound on inventory level.

In reality, uncertainty can sometimes make inventory violations inevitable, because trip delays and vessel unavailability disrupt the implementation of the solutions previously obtained. In such cases, we can still enforce constraints (4.23)–(4.25), with a small modification: (4.23) are enforced for every order k of third-party production node j except its first order in the horizon, and (4.24), (4.25) are enforced for time t greater than a threshold value $\max_{j'} \tau_{j'j}$

Finally, when all the vessels have the same capacity and availability profile, the symmetry breaking constraints shown below can be added,

$$\sum_j W^{L}_{i',vc,jt} \leq 1 - \bar{X}^{L}_{i,vc,t+1} \quad \forall i, i' > i, t \tag{4.26}$$

where vessel i' is not allowed to move out from the vessel center at time t, if vessel $i < i'$ is not rented in time t+1 ($\bar{X}^{L}_{i,vc,t+1} = 1$). Thus, if two vessels are available for renting and only one is needed, the one with a smaller index is rented.

4.3. Uncertainty and Stochastic Simulations

In practice, there are different sources of uncertainty which can make the original solution suboptimal or even infeasible as the horizon is rolled forward. To understand how solution quality can deteriorate, we study the following uncertainties: vessel availability, trip delays, pick-up window information, and consumption/production rates. We first discuss how they affect the problem, and then describe how we incorporate them in the stochastic simulation. When needed, we use a plus sign superscript to denote the parameters in the current iteration (e.g., ρ^{+}_{jmt}), and a minus sign for the parameters used in the previous iteration (e.g., ρ^{-}_{jmt}). The algorithms for stochastic simulations are presented in the Appendix G.

4.3.1. Vessel Availability in Long-term Mode

Before a vessel is rented, the decision maker needs to "call" the vessel company within what we refer to as the "reservation window", to check availability and make the reservation. We use λ^{LA} and λ^{LB} to denote the start and the end of the reservation window. In other words, the availability of vessel i is checked, if it needs to be rented in the open-loop solution (i.e., $\sum_{j,\lambda^{LA}\le t\le\lambda^{LB}} W^{L}_{i,vc,j,t} = 1$), as shown in Figure 4.5. The probability of a vessel being available at time t is $\varepsilon_L(t)$. Generally, the earlier a call is made, the more likely a vessel is to be available, i.e., $\varepsilon_L(t)$ is a non-decreasing function of t. If a vessel is not available, we get feedback about the time when it will become available, add constraints on availability, and re-solve the model.

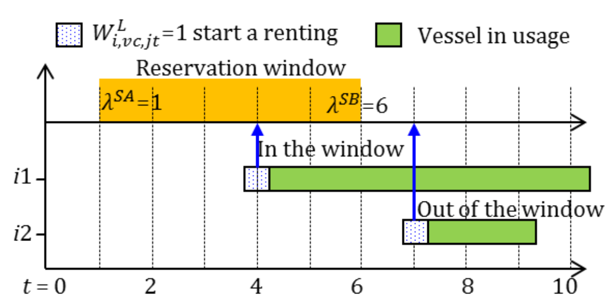


Figure 4.5. Availability of vessels is checked if the solution of the model includes the start of a long-term renting within the reservation window ($t \in [\lambda^{LA}, \lambda^{LB}]$).

In the stochastic simulation (Algorithm 4.1), $\mathbf{I}^R$ denotes the set of vessels in long-term mode that are still at the vessel center, but already reserved, and thus the availability of vessels in $\mathbf{I}^R$ is not checked. Stochastic parameters subject to uniform distributions are generated to determine the availability of vessels. We assume that the probability that a vessel is available is a linear function of t. To model the vessel availability, we introduce the following:

(a) Index $n \in \mathbf{N} = \{1, \dots, N\}$ to denote the number of unreserved vessels;

(b) Parameter $nmax$ to denote the maximum number of unreserved vessels that are possible to start a long-term renting within the reservation window;

(c) Parameter δ_n^L to denote the time that the n^{th} vessel becomes available;

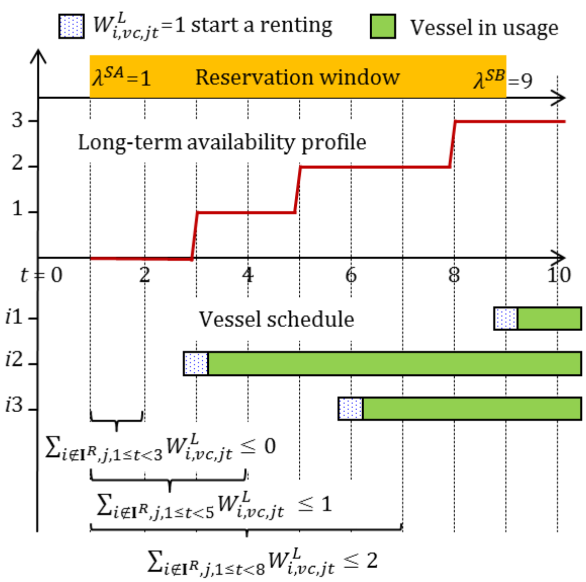


Figure 4.6. Constraining the number of unreserved vessels in long-term mode according to the availability profile.

If the n^{th} unreserved vessel is available at t, such that $\sum_{i \notin \mathbf{I}^R, j} W_{i,vc,j,t}^L = 1$, then $\delta_n^L = t$; otherwise, the model is re-solved with updated availability information. The number of unreserved vessels is constrained as follows,

$$\sum_{i \notin \mathbf{I}^R, j,\, \lambda^{LA} \le t < \delta_n^L} W_{i,vc,jt}^L \le n - 1 \quad \forall n \le nmax \tag{4.27}$$

which means that before the n^{th} vessel becomes available, and after the earliest long-term reservation time, at most n-1 vessels can be moved out of the vessel center (see Figure 4.6).

4.3.2. Vessel Availability in Short-term Mode

Vessels rented in short-term mode should be reserved during a short-term reservation window; we use λ^{SA} and λ^{SB} to denote the start and the end of the reservation window. If the open-loop solution includes a trip (by a short-term rental) yet to be reserved during the window, a "call" is made to check availability. If there is availability, the reservation is made and the horizon is rolled forward; otherwise, constraints based on the received feedback are added and the model is re-solved. The overall procedure is similar to the one for long-term rentals, but the short-term rental availability depends on the location of the trip, as described next.

Arcs (trips) are grouped into clusters, denoted by $l \in \mathbf{L}$, according to their origins and destinations. A cluster includes arcs that are close to each other. We assume that the number of available vessels to serve different trips in a cluster is the same for all trips in that cluster. As in §4.1, index $n \in \mathbf{N} = \{1, \dots, N\}$ denotes the number of unreserved trips, and the n^{th} vessel to serve a trip in cluster l becomes available at $t = \delta^{S}_{ln}$, where the stochastic parameter δ^{S}_{ln} evolves as the horizon is rolled (Algorithm 4.2). Based on the values of variables $W^{S}_{jj't}$, we know the "desired" renting time for the n^{th} trip in each cluster. If the desired renting time is before the availability time, the model is re-solved (Algorithm 4.3).

The number of unreserved trips is constrained as follows,

$$\sum_{\lambda^{SA} \le t < \delta^{S}_{ln}, (j,j') \in \mathbf{A}_l \setminus \mathbf{A}^{R}_{t},} W^{S}_{jj't} \le n - 1 \quad \forall n, l \tag{4.28}$$

where $\mathbf{A}_l$ denotes the set of trips in cluster l, and $\mathbf{A}^{R}_{t}$ denotes the set of trips that start at time t and are already reserved. Thus, constraints (4.28) require that for each cluster l, after the start of the short-term reservation window λ^{SA} and before the vessel to serve the n^{th} trip becomes available, at most n-1 unreserved trips are allowed to start.

4.3.3. Trip Delay

Scheduled trips may be delayed due to three reasons. First, the newly rented vessels may arrive at their destination node late. Second, travel time of an on-going trip may vary, typically because of weather-related reasons. Since longer trips have potential for larger delays, this second type of delay is roughly proportional to trip duration. Third, unexpected events in pick-ups or deliveries can extend any trip by 1 or 2 days. Trip delays are simulated using Algorithm 4.4.

4.3.4. Pick-up Window Specifications

Each order from a third-party production node should be picked up within a window. The pick-up windows starting before $t = \lambda^{PU}$ are deterministically known and are usually 2 to 3 days wide; normally, $\lambda^{PU} = 30$ days. For the remaining orders (with $t > \lambda^{PU}$), no window information is given, so the start times of the windows are estimated based on order frequency. Also, the width of these windows is assumed to be 10 days. The information on these windows becomes deterministically known when the estimated start time is equal to $t = \lambda^{PU}$ (this process reflects industrial practice). The order amount, which in the general case will be stochastic, is assumed to be deterministic here. The simulation of pick-up window specifications is carried out using Algorithm 4.5.

4.3.5. Consumption/Production Rate

The actual consumption/production rate in the period just before $t = 0$ may differ from its forecast, and the future forecast rate may change as the horizon is rolled forward. We assume that these differences follow a normal distribution. For example, we assume that the change of future forecast is $(\rho_{jmt}^{+} - \rho_{jmt}^{-})/\rho_{jmt}^{-} \sim \mathcal{N}(0, \sigma_{FF})$, where $\rho_{jmt}^{-}/\rho_{jmt}^{+}$ denotes the old/new forecast rate, and σ_{FF} denotes the standard deviation of the percentage change in the forecast rate. Algorithms 4.6 and 4.7 are used to simulate the actual production/consumption rates and forecast changes.

4.4. Reoptimization Framework

We reoptimize at a given frequency, typically once a day. At each stage, we observe uncertainty, update parameters and constraints, and re-solve the optimization model. We repeat D-1 times, so D open-loop solutions with different initial time are obtained. We use date $d \in \mathbf{D} = \{1,2,\dots,D\}$ to denote the absolute time each open-loop solution was obtained, and period/point t to denote the relative time in each solution (i.e., t = 0 is the initial time of each MIP model).

We use two MIP models: **M1**, without availability constraints (equations (4.4)–(4.26)); and **M2**, with availability constraints (equations (4.4)–(4.28)). Before solving either model, we fix the corresponding variables according to long-/short-term renting and long-term returning decisions that have been made previously (see Algorithm 4.8).

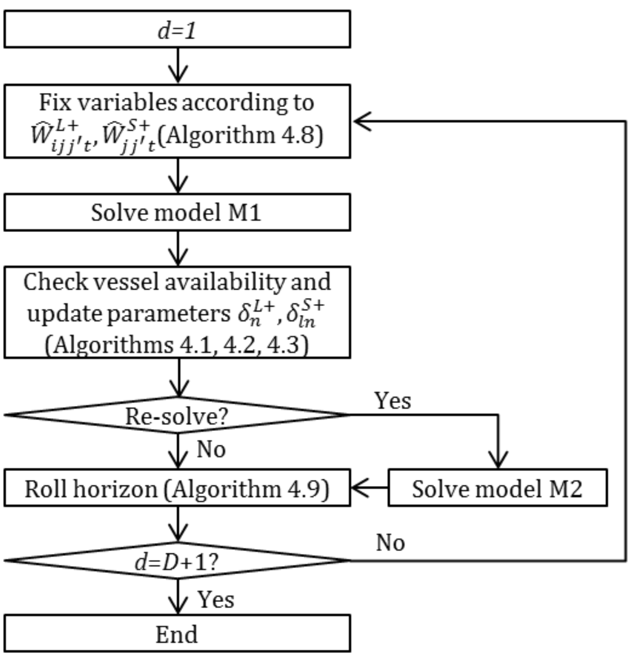


Figure 4.7. Flowchart of the reoptimization algorithm.

To obtain each open-loop solution, we first solve model **M1**, and then simulate the “call” to check vessel availability (for both long- and short-term modes). If the desired vessels are available, the reservations are made and the horizon is rolled forward. Otherwise, if any of the desired vessels is unavailable, we evaluate what the decision maker would do by adding constraints (4.27) and

(4.28), and solving **M2**. Based on the solution of **M2**, we make the new reservations, and roll the horizon. The overall algorithm is summarized in Figure 4.7, and the detailed procedure to roll the horizon is given in Algorithm 4.9.

We use parenthesis to denote the variable values from the open-loop solution at date d (e.g., $C_t^{MH}(d)$ denotes the material holding cost during period t in the open-loop solution obtained at date d). We use $C_t(d)$ to denote the overall cost during period t from the open-loop solution at date d, which is calculated as follows,

$$C_t(d) = C_t^{MH}(d) + C_t^{OF}(d) + C_t^{UF}(d) + C_t^{FT}(d) + C_t^{VT}(d) + C_t^{FL}(d) + C_t^{EL}(d) + C_t^{S}(d) \quad (4.29)$$

We also introduce the *estimated cost* at date d, denoted by $C^{ID}(d)$, which has two components: (1) the actual cost $CA^{ID}(d)$, between 0 and d-1, based on already implemented decisions; and (2) the forecast cost $CB^{ID}(d)$, between d and D, based on future decisions in the open-loop solution at d.

$$CA^{ID}(d) = CA^{ID}(d-1) + C_1(d-1) \quad (4.30)$$

$$CB^{ID}(d) = \sum_{t \le D-d+1} C_t(d) \quad (4.31)$$

$$C^{ID}(d) = CA^{ID}(d) + CB^{ID}(d) \quad (4.32)$$

Therefore, $C^{ID}(1)$ denotes the first open-loop cost, while $C^{ID}(D)$ denotes the closed-loop cost after the horizon is rolled D-1 times. The evolution of estimated cost is illustrated in Figure 4.8. In §4.6, we will discuss how the evolution of the estimated cost helps us understand the effect of reoptimization.

4.5. Policy Analysis

As new information becomes available and uncertainty is observed, previously obtained open-loop solutions become suboptimal or even infeasible. Interestingly, solutions that appear to be good when computed, result in poor implemented solutions, which leads us to study four policies that can be adopted, on top of the optimization models solved at each iteration, to improve the quality of the implemented schedule:

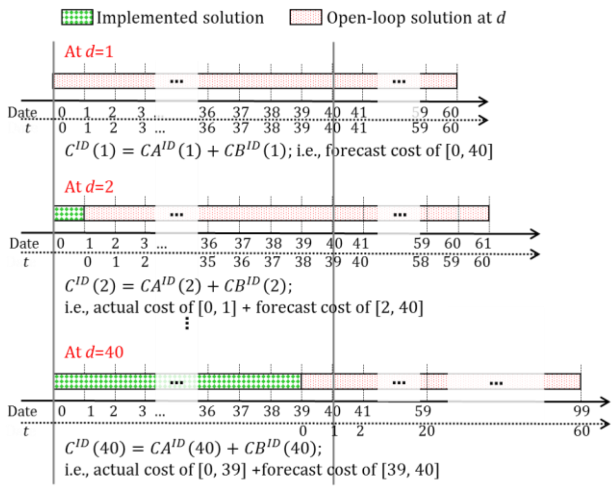


Figure 4.8. The estimated cost $C^{ID}(d)$, as the horizon is rolled, from d=1 to d=D=40, using an open-loop solution with η=60 (i.e., $t \in [0,60]$ in each model).

As new information becomes available and uncertainty is observed, previously obtained open-loop solutions become suboptimal or even infeasible. Interestingly, solutions that appear to be good when computed, result in poor implemented solutions, which leads us to study four policies that can be adopted, on top of the optimization models solved at each iteration, to improve the quality of the implemented schedule:

(a) Adjusting the start and end time of (long- and short-term) reservation windows;

(b) Placing restrictions on the minimum number of vessels rented in long-term mode;

(c) Adding a preference for early pick-up during a window; and

(d) Modifying start time and length of pick-up windows.

4.5.1. Reservation Windows

The start and the end times of the windows are adjustable parameters by the decision maker. Reserving early makes it hard to adjust the schedule as uncertainty is observed, while attempting to reserve late can lead to unavailability. If there is limited uncertainty, early reservations are expected to be favorable. Similarly, if vessel availability decreases fast with time, then early reservations are also expected to lead to better solutions. Thus, given the uncertainty in the system, our goal is to study how reservation windows should be chosen.

4.5.2. Vessel Constraints

In general, optimization over a short horizon tends to yield myopic solutions where, for example, few vessels are rented in long-term mode because, among others, (1) many trips are assigned to be performed by short-term vessels which are then found to be unavailable (unlike long-term rentals); and (2) idle long-term rentals can be used to react to last minute delays. Accordingly, we study how adding a lower bound, *vmin*, on the number of vessels rented in long-term mode (modeled through vessels staying at the vessel center) can be beneficial:

$$\sum_{i} \bar{X}^{L}_{i,vc,t} \leq |\mathbf{I}| - vmin \quad \forall t \tag{4.33}$$

4.5.3. Early Pick-up

If the pick-up of an order can be scheduled at any time during the pick-up window without affecting the objective value, the optimization can arbitrarily allocate this pick-up to any time in that window. However, when uncertainty is considered, it would be favorable to schedule the pick-up early in this window to leave some room for trip delays, thereby leading to better closed-loop solutions. The preference of early pick-up is modeled using a small penalty term:

$$C^{EP}_{t} = \sum_{i,j \in \mathbf{J}^{TP},j'} \pi^{EP}_{jt} W^{L}_{ijj't} + \sum_{j \in \mathbf{J}^{TP},j'} \pi^{EP}_{jt} W^{S}_{jj't} \quad \forall t \tag{4.34}$$

where parameter $\pi^{EP}_{j,t}$ is calculated as follows,

$$\pi^{EP}_{jt} = \sum_{k \in \mathbf{K}_j} \theta_{jkt} \cdot \beta \cdot \left(t - \left\lceil \sigma^{OS}_{jk} / \delta \right\rceil\right) \tag{4.35}$$

and β is a penalty parameter. In equation (4.35), if t is within the window of order k from node j, the earlier t is, the smaller parameter π_{jt}^{EP} becomes. After adding term C_t^{EP}, objective function (4.12), is modified as follows,

$$\min C^{ALL} = \sum_t (C_t^{MH} + C_t^{OF} + C_t^{UF} + C_t^{FT} + C_t^{VT} + +C_t^{FL} + C_t^{EL} + C_t^{S} + C_t^{EP}) \tag{4.36}$$

To implement this preference, we first solve the original model, and then fix the vessel-trip assignment based on the solution of the original model and re-solve the model with constraints (4.34)-(4.36).

4.5.4. Pick-up Windows

Pick-ups from third-party production nodes are expected to be carried out within the pick-up windows. Due to either the limited number of vessels or the uncertainty in the system, it is sometimes impossible to perform all pick-ups within these windows, which results in a penalty. To address this, we study three remedies, which the decision-maker can potentially negotiate:

(a) Earlier specification of pick-up windows;

(b) Longer pick-up windows;

(c) *Soft* pick-up windows.

When soft pick-up windows are introduced, late/early pick-ups just outside the windows are penalized with a comparatively small overflow/underflow cost. We use σ^{SW} to denote the number of periods in a soft pick-up window. If an order cannot be picked up within this soft window, the original large penalty is used (see Figure 4.9). We study different settings: (1) no soft pick-up windows ($\sigma^{SW} = 0$); (2) soft pick-up windows with $\sigma^{SW} = 1$; (3) soft pick-up windows with $\sigma^{SW} = 2$. For (2) and (3), we study different penalties to assess what penalties would be acceptable.

4.6. Case Study

We consider an instance with 2 third-party production nodes, 1 owned production node, 2 consumption nodes, and 1 material. There are 7 vessels for long-term renting, which are enough to generate the optimal solution. The time step is 1 day, and the planning horizon is 60 days. To acquire one closed-loop solution, 40 open-loop solutions are obtained, i.e., $D = 40$.

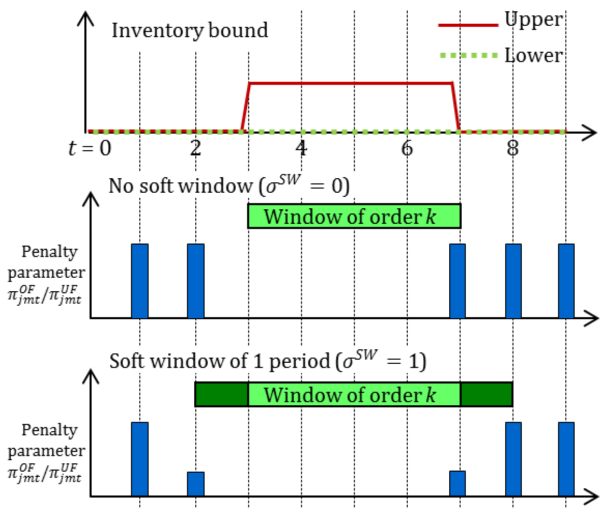


Figure 4.9. Penalty modification for soft pick-up window.

All models and algorithms were implemented in AIMMS 3.13 and solved using CPLEX 12.6 on a machine with two 2.26 GHz Intel Xeon E5520 processors and 16GB RAM running Windows 8. The solution time limit was 210 seconds for each MIP model. The optimality gap was, on average, 0.5%.

4.6.1. Effect of Short-term Renting and Reoptimization

We study four different solutions: (1) open-loop solution at d=1 without short-term renting; (2) closed-loop solution without short-term renting; (3) open-loop solution at d=1 with short-term renting; and (4) closed-loop solution with short-term renting. No uncertainty is considered.

The four solutions are shown in Figures 4.10, 4.11, where V1-V7 represent vessels in long-term mode, while VS represents vessels in short-term mode. Estimated costs $C^{ID}(d)$ are shown in Table 4.1. Incorporating short-term renting can reduce the distribution cost by 15% in the open-loop solution, and 8% in the closed-loop solution. We also see in Figure 4.11 that when short-term renting is allowed, more than 20% of the pick-ups from production nodes are made by vessels rented in short-term mode.

Table 4.1. Estimated costs $C^{ID}(d)$ of the four solutions.

	Without short-term	With short-term
Open-loop	180,000	153,750
Closed-loop	188,950	174,750

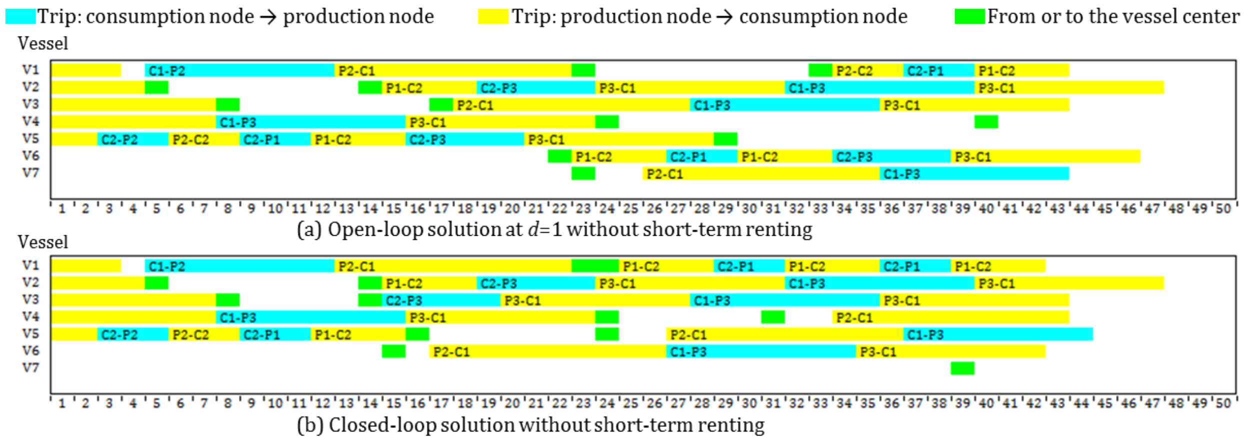


Figure 4.10. Open- and closed-loop solutions, without short-term renting.

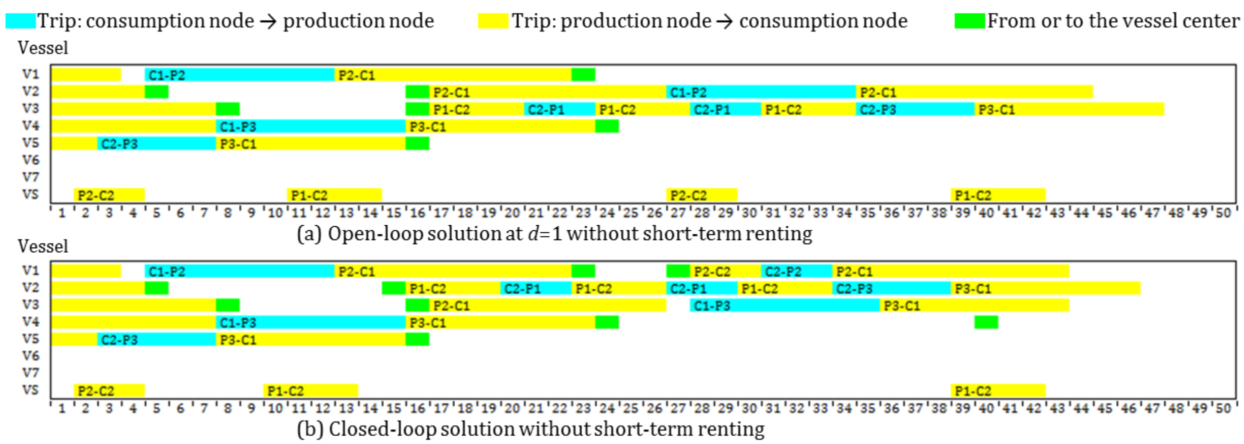


Figure 4.11. Open- and closed-loop solutions, with short-term renting.

We also see that closed-loop cost is 5% to 14% higher than the corresponding open loop cost at d = 1. This is primarily due to the effect of the finite horizon problem we solve. Specifically, (1) vessels in long-term mode are rented for longer periods in the closed-loop solution, considering the deliveries beyond the horizon of the model solved at d=1; and (2) in the open-loop solution at d = 1, the inventory level of the owned production node (P3 in Figures 4.10, 4.11) at the end of horizon is high to minimize the distribution cost, but implementing this solution would lead to inventory violations beyond the horizon, which leads to more pick-ups from this node in the closed-loop solution.

These results suggest that the addition of short-term renting is essential to reduce the distribution cost. Thus, short-term renting is incorporated in all the following studies. Moreover, we verify that even in the deterministic case, the closed-loop solution is different from the initial open-loop solution, which suggests that methods to obtain high quality closed-loop solutions have to be studied even for the deterministic case.

4.6.2. Effect of Uncertainty

The following five problems are studied: (1) deterministic (all pick-up windows are specified); (2) problem under uncertain vessel availability (§4.3.1, §4.3.2); (3) problem with trip-delays (§4.3.3); (4) problem with uncertain pick-up window specifications (§4.3.4); and (5) problem under consumption and production rate uncertainty (§4.3.5). We generate 60 samples for each problem.

Some results are shown in Figure 4.12, where $C^{ID}(d)$ for a typical but specific sample and the average of $C^{ID}(d)$ over samples are given as functions of *d*. As expected, the fluctuation for the specific sample is large, while the fluctuation of the average value of $C^{ID}(d)$ decreases. The open-loop cost, $C^{ID}(1)$, is the same for all the problems except problem (4), in which the length of pick-up windows starting after $t = \lambda^{PU}$ are increased to 10 days at *d*=1, leading to a lower open-loop cost. Nevertheless, the closed-loop cost, $C^{ID}(D)$, is the focus of our study.

Note that $C^{ID}(D)$ for one sample in Figure 4.12 has large fluctuations because there are multiple optimal solutions which may lead to changes in $C^{ID}(d)$. For example, in the deterministic problem solved at d = 16, there are seven pick-ups from the owned production node; three of them start in [16,40], which are used to calculate $C^{ID}(d)$, and four after d = 40. However, in the open-loop solution obtained at *d*=17, two of the four pick-ups beyond d = 40 are moved earlier (arbitrarily due to solution symmetry), thereby leading to a solution with five (rather than three) out of the seven pick-ups scheduled before d = 40, which leads to an increase of $C^{ID}(d)$. Note that $C^{ID}(d)$ profiles become flat as d approaches 40 because decisions for small t in the corresponding open-loop solutions do not change.

As expected, the deterministic problem has the lowest closed-loop cost. On average, the cost of the problem under consumption/production rate uncertainty does not increase significantly compared to the deterministic problem, because the variations cancel out in the long-run, though they do lead to slightly different decisions. The problem under pick-up window specification uncertainty has a closed-loop cost 3% higher than the deterministic problem. Incorporation of vessel availability and trip duration uncertainty increases the closed-loop cost by 14% and 10%, respectively, compared to the deterministic case.

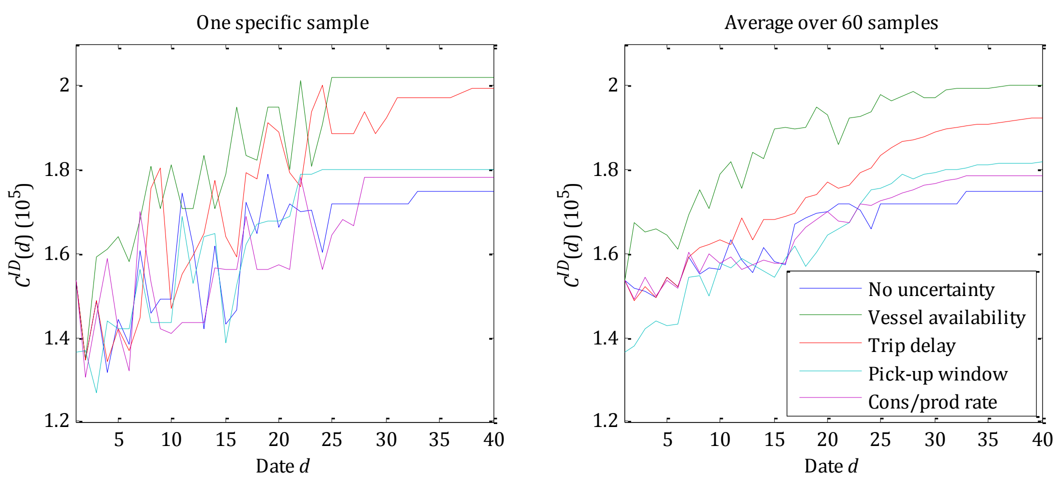


Figure 4.12. Profiles of $C^{ID}(d)$, when different sources of uncertainty are incorporated.

4.6.3. Effect of Policies

As discussed in §4.5, adopting different policies may lead to different open- and closed-loop solutions, especially in the presence of uncertainty. Accordingly, we study how different policies affect closed-loop solutions. Policies related to pick-up windows are studied in the next subsection, since they need to be negotiated with the third-party production nodes.

For the three groups of policies described above, 13 cases are studied (see Table 4.2). The best reservation window parameters are obtained using the results of cases 1-6. The decision to adopt vessel constraints and the corresponding parameters are studied through cases 7-11, and the preference for early pick-ups is studied in cases 12-13. All sources of uncertainty are incorporated in the reoptimization. We generate 60 samples for each case and compare the different policies using the mean of the closed-loop cost.

The closed-loop cost is a random variable. The closed-loop cost of 60 samples and the corresponding histogram are shown in Figure 4.13, along with a fitted normal distribution. We can see that the distribution of closed-loop cost is similar to the fitted normal distribution. The similarity is observed for all the other cases as well.

Table 4.2. Characteristics of 13 cases.

Policy to study	Case	Reservation windows		Vessel constraints (minimum number)	Preference of early pick-ups
		Long-term	Short-term		
Reservation windows	1	[14,21]	[2,10]	No	No
	2	[14,21]	[2,5]		
	3	[14,21]	[5,10]		
	4	[7,14]	[2,10]		
	5	[7,14]	[2,5]		
	6	[7,14]	[5,10]		
Vessel constraints	7	Best from cases 1-6		No	No
	8			Yes (3)	
	9			Yes (4)	
	10			Yes (5)	
	11			Yes (6)	
Preference of early pick-ups	12	Best from cases 1-6		Best from cases 7-11	No
	13				Yes

If we view all closed-loop solutions for each case as a population, the expected value of random variables in 2 populations (i.e., the expected closed-loop cost from solutions in 2 cases) can be compared (Wonnacott and Wonnacott, 1990), as shown in the following equation,

$$(\mu_1 - \mu_2)_\alpha = (\overline{X_1} - \overline{X_2}) \pm t(\frac{1-\alpha}{2}) \sqrt{(\frac{\sum_m (X_{1,m} - \overline{X_1})^2 + \sum_m (X_{2,m} - \overline{X_2})^2}{n_1 + n_2 - 2}) \cdot (\frac{1}{n_1} + \frac{1}{n_2})} \quad (4.37)$$

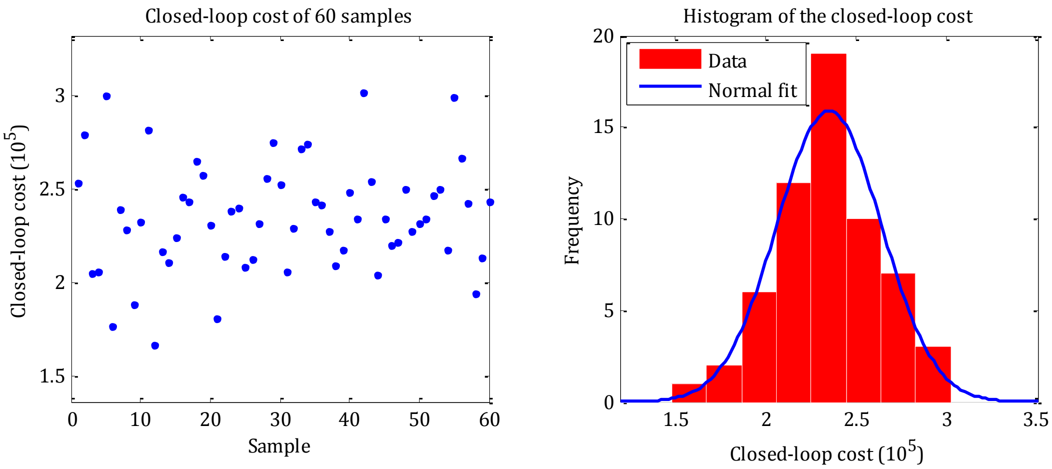


Figure 4.13. Results of the closed-loop cost in case 2.

where $(\mu_1 - \mu_2)_\alpha$ is the range of the difference of expected values at confidence level α, $t(x)$ is the value above which the area of probability density function is *x* for student's t-distribution. If we use population 1 as reference, n_1 denotes the sample size, $\overline{X_1}$ denotes the sample mean, and $X_{1,m}$ denotes the random variable value (closed-loop cost) of sample *m*. The number of degrees of freedom for $t(\cdot)$ is $n_1 + n_2 - 2$. Based on equation (4.37), if the sample means of closed-loop cost in case 1 and case 2 satisfy $\overline{X_1} > \overline{X_2}$, we can use equation (4.38) below to calculate the confidence level, α, that case 1 leads to higher closed-loop cost than case 2 ($\mu_1 > \mu_2$).

$$\alpha = 1 - t^{-1}\left\{(\overline{X_1} - \overline{X_2})\sqrt{\frac{n(n-1)}{\sum_m\left(X_{1,m} - \overline{X_1}\right)^2 + \sum_m\left(X_{2,m} - \overline{X_2}\right)^2}}\right\} \tag{4.38}$$

where $n = n_1 = n_2$ denotes the number of samples we have for each case, $t^{-1}(\cdot)$ is the inverse function of $t(\cdot)$, and the number of degrees of freedom is 2(*n*-1). The results and statistics are shown in Tables 4.3-4.5, respectively for the three groups of policies. We make the following remarks:

(a) Comparing cases 1-6 (Table 4.3), we observe that case 4 (using [7,14] in long-term mode and [2,10] in short-term mode) has the lowest sample mean of closed-loop cost. Thus, for the uncertainty used in our simulations, the closed-loop cost can be minimized by using a relatively late long-term reservation window and a wide short-term reservation window. Given the confidence levels in Table 4.3, we conclude that the policies in case 4 outperform the others.

(b) The comparison of cases 7-11 (Table 4.4) suggests that adding vessel constraints leads to lower cost. Case 10, where the minimum number of vessels (*vmin*) is 5, has the lowest sample mean.

(c) The comparison of cases 12 and 13 (Table 4.5) suggests that early pick-ups lead to lower closed-loop cost, with a confidence level of 80%.

Based on these 13 cases, we choose the reservation windows in long- and short-term modes to be [7,14] and [2,10], respectively; use vessel constraints with at least 5 vessels; and prefer early pick-ups. We verified that making any changes in the policies used in case 13 does not lead to improvements.

4.6.4. Effect of Policies related to Pick-up Windows

The policies related to pick-up windows are studied separately, since they have to be negotiated with third-party production nodes. We test multiple cases, using different λ^{PU}, window length ($\sigma_{jk}^{OE} - \sigma_{jk}^{OS}$), and σ^{SW}. For each case, we generate 60 samples; sample means of the closed-loop cost, as well as standard deviations, are summarized in Table 4.6. Soft window penalty is the ratio of the inventory violation penalty during soft windows over the same penalty during other periods. Adding 1- and 2-day soft windows can lower the cost by 7% and 13%, respectively (see cases 14, 16 and 17). Using windows between 4 and 5 days leads to about the same as using windows between 2 and 3 days combined with 2-day soft windows (comparing cases 15 and 17). Using larger λ^{PU} does not change the closed-loop cost much, because the information farther in the horizon plays a small role (see cases 14 and 18).

Different soft window penalties are also studied, and the results are shown in Figure 4.14. We observe that as the soft window penalty increases to 0.7, the sample mean of the closed-loop cost becomes similar to the original case (no soft windows, case 14). Note that the frequency of pick-ups during the soft windows increases as (1) the probability of trip delays increases, (2) the probability of vessels being available decreases, and (3) pick-up windows from several production nodes have similar start times. The decision maker can use these insights to negotiate soft window penalties.

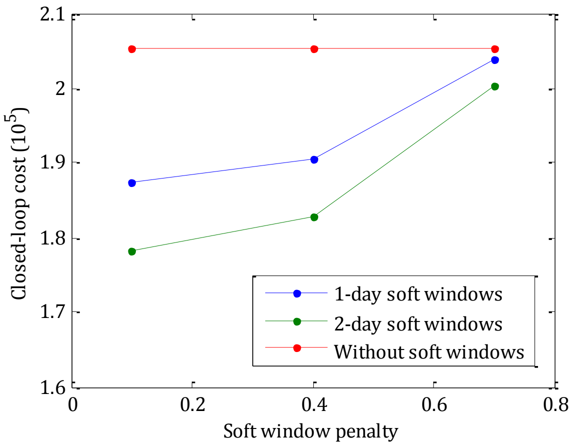


Figure 4.14. Sample mean of the closed-loop cost for different soft window length and penalties.

Table 4.3. Statistics of cases used to study reservation window paramters. The table includes the sample mean of closed-loop cost ($\overline{C^{ID}}(D)$), sample standard deviation (SD) of the closed-loop cost, and the confidence level (α) that the case with the smallest sample mean has a smaller expected value than the case in the column. Best denotes the case in that column has the smallest sample mean.

Case	1	2	3	4	5	6
$\overline{C^{ID}}(D)$ (10^5)	2.31	2.35	2.47	2.23	2.41	2.44
SD (10^4)	2.27	2.89	2.28	1.50	2.24	1.53
α (%)	98.86	99.74	99.99	Best	99.99	99.99

Table 4.4. Statistics of cases with different vessel constraints. The table includes the sample mean of closed-loop cost ($\overline{C^{ID}}(D)$), sample standard deviation (SD) of the closed-loop cost, and the confidence level (α) that the case with the smallest sample mean has a smaller expected value than the case in the column. Best denotes the case in that column has the smallest sample mean.

Case	7	8	9	10	11
$\overline{C^{ID}}(D)$ (10^5)	2.23	2.21	2.11	2.08	2.12
SD (10^4)	1.50	1.56	1.69	2.19	2.16
α (%)	99.99	99.98	74.16	Best	80.96

Table 4.5. Statistics of cases studying whether to use the preference of early pick-ups. The table includes the sample mean of closed-loop cost ($\overline{C^{ID}}(D)$), sample standard deviation (SD) of the closed-loop cost, and the confidence level (α) that the case with the smallest sample mean has a smaller expected value than the case in the column. Best denotes the case in that column has the smallest sample mean.

Case	12	13
$\overline{C^{ID}}(D)$ (10^5)	2.08	2.05
SD (10^4)	2.19	1.67
α (%)	80.16	Best

Table 4.6. Statistics of cases with different pick-up windows.

Case	14	15	16	17	18
Window specifying time	30	30	30	30	45
Window length	[2,3]	[4,5]	[2,3]	[2,3]	[2,3]
Soft window length	0	0	1	2	0
Soft window penalty			0.1	0.1	
$\overline{C^{ID}}(D)$ (10^5)	2.05	1.78	1.90	1.78	2.04
SD (10^4)	1.67	1.64	1.57	1.59	1.73

4.7. Conclusions

We developed a framework for reoptimization in maritime inventory routing under uncertainty. The proposed framework, which includes solving MIP models and implementing stochastic simulations, can be generalized to handle any inventory routing problem. Specifically, we developed (1) a discrete-time MIP model considering vessels in long- and short-term renting modes, as well as owned and third-party production nodes; (2) stochastic simulations to account for uncertainty sources that appear in practice; and (3) a reoptimization algorithm integrating the MIP model and stochastic simulations. Since the quality of the implemented solution depends heavily on a number of policies, we studied the effect of different policy parameters.

Using a number of case studies, we first showed that the open- and closed-loop problems are very different: even when no uncertainty is present, the closed-loop cost could be 5-15% higher than the open-loop cost. The average closed-loop cost in the presence of all uncertainty sources was nearly 30% higher than that in the deterministic case. Uncertainty of trip duration and vessel availability increased the closed-loop cost significantly. We also discussed how to identify policy parameters (including reservation windows, constraints on the number of rented vessels, preference for early pick-ups, and different types of pick-up windows) that result in high quality implemented solutions.

4.8. Notation

Indices/Sets

$d \in \mathbf{D}$ dates (absolute time)

$i \in \mathbf{I}$	vessels
$j \in \mathbf{J}$	nodes in the SC network, including vessel center (*vc*)
$(j, j') \in \mathbf{A} \subseteq \mathbf{J} \times \mathbf{J}$	arcs in the SC network
$k \in \mathbf{K}_j$	orders of third-party production node *j*
$l \in \mathbf{L}$	clusters
$m \in \mathbf{M}$	materials
$n \in \mathbf{N}$	number of unreserved vessels/trips
$t \in \mathbf{T}$	time points or periods

Subsets

$\mathbf{A}_l$	arcs that are within in cluster *l*
$\mathbf{A}_t^R$	arcs (trips) that are already reserved at time *t* (in short-term mode)
$\mathbf{I}^R$	vessels (in long-term mode) that are still at the vessel center, but already reserved
$\mathbf{J}^P/\mathbf{J}^C$	production/consumption nodes
$\mathbf{J}^{TP}/\mathbf{J}^{OP}$	third-party/owned production nodes

Binary Variables

$W_{ijj't}^L$	=1 if vessel i starts a trip from *j* to *j'* at time point *t*
$W_{jj't}^S$	=1 if a vessel in short-term mode starts a trip from *j* to *j'* at time point *t*
$\bar{X}_{ijt}^L$	=1 if vessel *i* is at node *j* during time period *t*
$\bar{Y}_{it}^L$	=1 if renting of vessel *i* is extended in period *t* beyond ϑ^L periods

Non-Negative Variables

$C^{ALL}/C_t^{OF}/C_t^{UF}$	total/overflow/underflow cost
$C_t^{MH}/C_t^{FT}/C_t^{VT}$	material holding/fixed transportation/variable transportation cost
$C_t^{FL}/C_t^{EL}/C_t^S$	fixed long-term/extended long-term/short-term renting cost
C_t^{EP}	penalty term for modeling the preference of early pick-ups

$F^{L}_{ijj'mt}$	material *m* in vessel *i* traveling from *j* to *j'* starting at *t* in long-term mode
$F^{S}_{jj'mt}$	material *m* in the short-term rental from *j* to *j'* starting at *t*
L_{jmt}	inventory level of material *m* at node *j* at time point *t*
$L^{OF}_{jmt}/L^{UF}_{jmt}$	overflow/underflow amount of material *m* of node *j* at time point *t*

Parameters

α	confidence level
β	penalty constant for modeling the preference of early pick-ups
$\gamma^{MAX}_{i}/\gamma^{MAX}$	capacity of vessel *i*/ vessels in short-term mode
$\gamma^{MIN}_{i}/\gamma^{MIN}$	minimum load on vessel *i*/ vessels in short-term mode when traveling from a production node to a consumption node
δ	time period length
δ^{LE}	earliest time when a vessel becomes available in long-term mode
$\delta^{L}_{n}/\delta^{S}_{in}$	time when the *n*th vessel becomes available in long-/short-term mode
ε_{L}	probability of availability in long-term mode
$\zeta^{MAX}_{jmt}/\zeta^{MIN}_{jmt}$	maximum/minimum level of material *m* in node *j* at time point *t*
η	planning horizon
θ_{jkt}	=1 if period *t* is in pick-up window *k* of third-party production node *j*
ϑ^{L}	minimum number of renting periods in long-term mode
$\lambda^{LA}/\lambda^{LB}/\lambda^{LR}$	earliest reservation/latest reservation/returning notice time in long-term mode
$\lambda^{SA}/\lambda^{SB}$	earliest/latest reservation time in short-term mode
λ^{PU}	time when a pick-up window becomes deterministically known
$\xi^{MAX}_{jj'}$	maximum allowable load along (*j,j'*)
$\pi^{MH}_{jm}/\pi^{OF}_{jmt}/\pi^{UF}_{jmt}$	material holding/overflow/underflow cost
$\pi^{FT}_{jj'}/\pi^{VT}_{jj'}$	fixed/variable transportation cost along (*j,j'*)

π_i^{FL}/π_i^{EL}	long-term renting cost for the minimum periods/ each extended period
π_{jt}^{EP}	penalty for modeling the preference of early pick-ups
$\pi_{jj'}^{S}$	short-term renting cost for traveling on (*j,j'*)
ρ_{jmt}	production (positive) or consumption (negative) rate of node *j* during period *t*
$\sigma_{jk}^{OS}/\sigma_{jk}^{OE}$	start/end time of the pick-up window of order *k* from third-party production node *j*
σ^{SW}	soft window length
$\tau_{jj'}$	traversal time of arc (*j,j'*)
φ_{jmk}	amount of material *m* in order *k* from third-party production node *j*
χ_{it}	=1 if period *t* is within the first ϑ^L periods of the current renting of vessel *i*
$C^{ID}(d)$	estimated cost at date *d*
$\hat{F}_{ijmt}^{L}$	amount of material *m* that is en route and will arrive at node *j* at *t* from vessel *i* in long-term mode
$\hat{F}_{j'jmt}^{S}$	amount of material *m* that is en route and will arrive at node *j* at *t* from production node *j'* in short-term mode
$\widehat{W}_{ijj't}^{L}$	=1 if vessel *i* should leave *j* for *j'* at *t*
$\widehat{W}_{jj't}^{S}$	=1 if a vessel in short-term mode should leave *j* for *j'* at *t*
$\hat{X}_{ijt}^{L}$	=1 if vessel *i* is at node *j* initially (*t*=0), or it is en route and will arrive at *j* at *t* (*t*>0)

Chapter 5

Terminal Constraints for Online Scheduling [5]

Production scheduling, as an optimization problem that arises in many sectors, has been widely studied (Drexl and Kimms, 1997; Proth, 2007; Verderame et al., 2010; Maravelias 2012b; Harjunkoski et al., 2014). In practice, the production process is subject to many factors of uncertainty, such as rush orders, yield losses, production delays, unit breakdowns, etc. After observing such an uncertain “trigger” event, the existing schedule needs to be modified for obtaining the (new) optimal schedule (Vieira et al., 2003; Ouelhadj and Petrovic, 2009). The other way to modify the schedule is to re-compute the schedule periodically in a moving horizon approach (Sand et al., 2000). Recently, it has been pointed out that despite uncertainty, online scheduling should be carried out periodically to consider the new information, such as new demand (Gupta and Maravelias, 2016; Gupta et al., 2016). At each iteration of online scheduling, an open-loop solution is obtained from solving an optimization problem; while the implemented scheduling solution, after observing the uncertainties (feedbacks), is called closed-loop solution. Most of the research efforts focus on how to account for the uncertainty when obtaining each open-loop solution by applying different optimization techniques, including robust optimization (Vin and Ierapetritou, 2001; Janak et al., 2007; Li and Ierapetritou, 2008; Lappas and Gounaris, 2016), stochastic programming (Bonfill et al., 2004), and fuzzy programming (Balasubramanian and Grossmann, 2003). For online scheduling, we should also aim to improve the closed-loop solution.

To obtain good closed-loop solutions, terminal constraints should be included in the scheduling model (Stadtler, 2000). In the optimal open-loop solution of a scheduling model without any terminal constraints, inventory tends to deteriorate at the end of horizon, so that the production,

---

[5] This chapter is modified from Dong and Maravelias, in preparation.

transition and inventory holding cost can be minimized (Lima et al., 2011). If (a part of) such an open-loop solution is implemented, the problem may become infeasible, or the closed-loop solution may be very costly, after the horizon is moved forward. This is because the inventory is depleted at the end of the previous horizon, and therefore the demand can hardly be satisfied. Thus, the scheduling model should include some terminal constraints to avoid the inventory depletion. This is similar to model predictive control (Mayne et al., 2000), in which the state variables of the last time are constrained in a terminal region. For scheduling problems, however, how to define the terminal region is not trivial.

Based upon the inventory management and SC literature, we can require the terminal inventory level be greater than a lead-time-based inventory threshold, which includes a buffer term named “safety stock” to address uncertainty (Eppen and Martin, 1988; Kreipl and Pinedo, 2004; You and Grossmann, 2008; Sana and Goyal, 2015). This threshold value is calculated from the statistics of the lead time of SC arcs and demand rate of SC nodes. On the other hand, based upon production scheduling literature, the terminal inventory levels can be required to be equal to the initial value at the start of the horizon (Baker, 1981; Shah et al., 1993), or to one of the values in a cyclic solution (Subramanian et al., 2012).

However, all of the aforementioned approaches neglect the relationship of inventory levels among materials. For instance, in a single-stage two-product problem, if the inventory level of one product is high, a low inventory level of the other can possibly be acceptable, because more resources can be allocated to produce the latter without leading to the stockout of the former. In this way, we can reduce the total inventory levels, and therefore save the inventory holding cost.

Accounting for the relationship among materials, we propose new types of terminal constraints on inventory levels for different network structures. These constraints are linear, and can be easily incorporated in any mixed integer programming (MIP) scheduling model. Theoretically, we prove that for deterministic problems of two types of networks, if the terminal inventory levels satisfy the proposed constraints, the scheduling model will be *recursively feasible*, which means that it will remain feasible after we move the horizon forward.

In Section 5.1, we present motivating examples and problem statement; the proposed framework to obtain the terminal constraints is shown in Section 5.2. Afterwards, we present the terminal constraints for different networks, including multi-stage single-product problems (Section 5.3), single-stage multi-product problems (Section 5.4), and multi-stage multi-product problems (Section 5.5); in these three sections, we consider a single machine in each stage. In Section 5.6, we generalize the terminal constraints to problems with multiple machines in each stage. In Section 5.7, we discuss how to apply the terminal constraints, and how to modify them considering uncertainty and periodic demand. In Section 5.8, we present computational results, using instances with and without uncertainty. Theoretical proofs are shown in the Appendix.

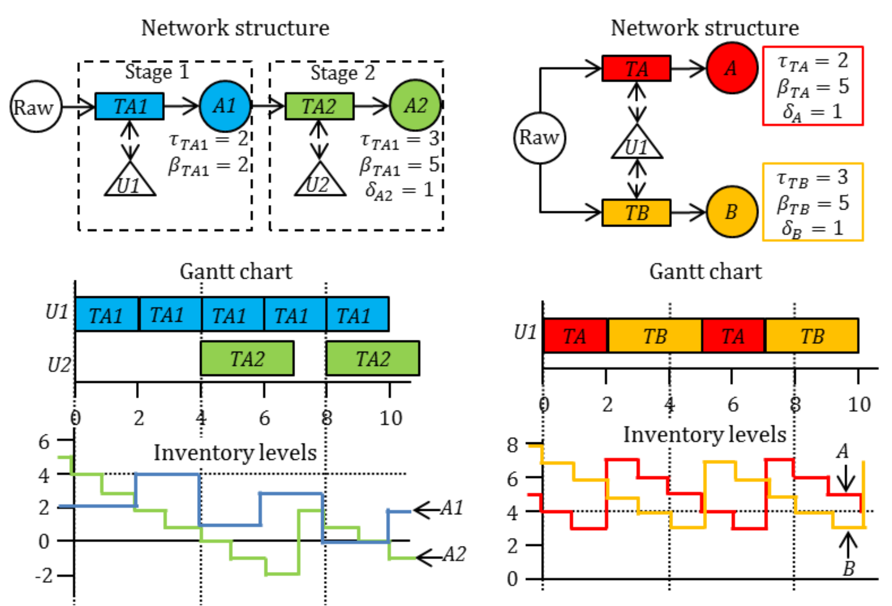
5.1. Background

5.1.1. Motivating Examples

The traditional lead-time-based threshold is calculated based on the statistics of the lead time and the demand (Eppen and Martin, 1988); if the problem is deterministic, the threshold is the lead time multiplied by the demand rate. However, simply requiring the terminal inventory level of each material to be greater than the threshold does not necessarily lead to good closed-loop solutions. This is illustrated through examples of deterministic problems with constant demands (Figure 5.1). We show that an initial inventory level equal to the threshold could be insufficiently low or unnecessarily high. Thus, the threshold approach cannot result in good terminal constraints.

In the two-stage single-product example (data given in Figure 5.1(a)), the lead time for the intermediate is the processing time in the first stage, and the lead time for the product is the summation of the processing times in both stages. The demand is 1 (ton/period), the lead time is 2 (periods) for $A1$, and 2+3=5 for $A2$, and thus the threshold is 2 (ton) for $A1$, and 5 for $A2$.However, the scheduling problem with such threshold as initial inventory would have stockout of product $A2$, because after the first batch of $TA1$ is finished, the inventory level of the intermediate is not enough to start $TA2$ immediately. This example shows that for multi-stage networks, the lead time is tricky to define; using values that are intuitively correct could result in stockout.

In the single-stage two-product example (data given in Figure 5.1(b)), the lead time is the summation of processing times for both tasks, since either can be processed after the other. The demand is 1 (ton/period), the lead time for both products is 2+3=5 (periods), and thus the threshold is 5 (ton). Following the thresholds, the initial inventory of A should be greater than 5, even when the initial inventory of B is fixed to be 8. However, this would lead to unnecessarily high inventory level; in fact, it would be sufficient to have an initial inventory of A being 2. In other words, when the initial inventory of one product is greater than the threshold, a lower value of the other could be acceptable.



(a) Two-stage single-product example (b) Single-stage two-product example

Figure 5.1. Network structure, Gantt chart and inventory levels of the motivating examples. Greek letters τ β and δ denote processing time, batch size and demand rate respectively. In the network structure, circles are for materials, triangles are for machines, and rectangles are for tasks.

These two simple examples show that adding traditional thresholds as terminal constraints cannot prevent stockout, or inventory holding cost may become high, because they neglect the relationship of materials in the network.

5.1.2. Problem Statement

The problem we consider is defined in terms of the following sets:

(a) $i \in \mathbf{I}$: tasks (or operations);

(b) $j \in \mathbf{J}$: machines (or units);

(c) $k \in \mathbf{K}$: stages;

(d) $m \in \mathbf{M}$: materials;

(e) $t \in \mathbf{T}$: time periods/points;

The horizon is divided into T uniform periods $t \in \{1, \dots, T\}$ with T+1 time points $t \in \{0,1, \dots, T\}$; period t starts at point t-1 and ends at t. The following subsets are also used to describe the problem:

(a) $\mathbf{I}_j \subseteq \mathbf{I}$: tasks that can be carried out in machine j;

(b) $\mathbf{I}_m^+/\mathbf{I}_m^- \subseteq \mathbf{I}$: tasks producing /consuming material m;

(c) $\mathbf{J}_i/\mathbf{J}_k \subseteq \mathbf{J}$: machines that can carry out task i /tasks in stage k;

(d) $\mathbf{M}^P \subseteq \mathbf{M}$: products;

We are given the following mappings:

(a) $i(m', k) \in \mathbf{I}$: the task in stage k to produce product m';

(b) $m(m', k) \in \mathbf{M}$: the material produced in stage k, which is used to produce product m';

Finally, we are given the following parameters:

(a) δ_m: *normalized demand* (constant demand in every period), if $m \in \mathbf{M}^P$; otherwise, $\delta_m = 0$.

(b) β_{ij}: batch size of task i in machine j;

(c) τ_{ij}: processing time of task *i* in machine *j*;

(d) α_{ij}: production cost of task *i* in machine *j*;

(e) π_m: inventory holding cost of material *m* for one period.

We make the following assumptions:

(a) Parameters δ_m β_{ij} and τ_{ij} are such that the demand can be fulfilled by production.

(b) Raw materials are always available, and therefore are not included in set **M**. For problems in which the inventory levels of raw materials are important, one dummy stage should be introduced before the first stage, and the arrival of the raw material should be viewed as the task of this dummy stage. By doing so, raw materials can be included in **M**, and the constraints that will be proposed still apply.

(c) The problem is deterministic. The problems with uncertainty are discussed in §5.7.

We use S_m to denote the terminal inventory level of material *m,* and $\boldsymbol{s}$ to denote the vector of terminal inventory levels; i.e., $\boldsymbol{s} = [S_{m1}, S_{m2}, \dots, S_{m|\mathbf{M}|}]^{\mathrm{T}}$. The scheduling problem is solved in a moving horizon approach; i.e., after the first period of the solution is implemented, the horizon (of the same length) is advanced forward, and the scheduling problem with new information is re-optimized. We want to study how to constrain the terminal inventory levels so that we can (1) ensure recursive feasibility, and (2) keep the inventory levels as low as possible.

5.2. Proposed Framework

5.2.1. Overall Approach

We first need to identify the *region of feasible terminal inventory levels,* denoted by $\mathbf{S}^{\mathrm{F}} \subseteq \mathbb{R}^{|\mathbf{M}|}$. Region $\mathbf{S}^{\mathrm{F}}$ is defined to be the largest set such that if $\boldsymbol{s} \in \mathbf{S}^{\mathrm{F}}$ is the value of the terminal inventory levels, the scheduling problem will remain feasible after the horizon is moved forward. To obtain region $\mathbf{S}^{\mathrm{F}}$, the assistance of a feasibility MIP model MF (given in §5.2.2) is needed, where $\boldsymbol{s}$ is used as a given parameter to denote the *initial* inventory levels. If model MF is feasible, a scheduling solution whose *terminal* inventory level is equal to the given $\boldsymbol{s}$ will lead to recursive feasibility (Figure 5.2(a)); i.e., $\boldsymbol{s} \in \mathbf{S}^{\mathrm{F}}$. After discretizing the inventory space, we check if a point $\boldsymbol{s}$ can lead to a feasible model MF, and thus decide if $\boldsymbol{s} \in \mathbf{S}^{\mathrm{F}}$. By repeatedly solving model MF with different values of $\boldsymbol{s}$, we obtain region $\mathbf{S}^{\mathrm{F}}$ (orange blocks in Figure 5.2(b)).

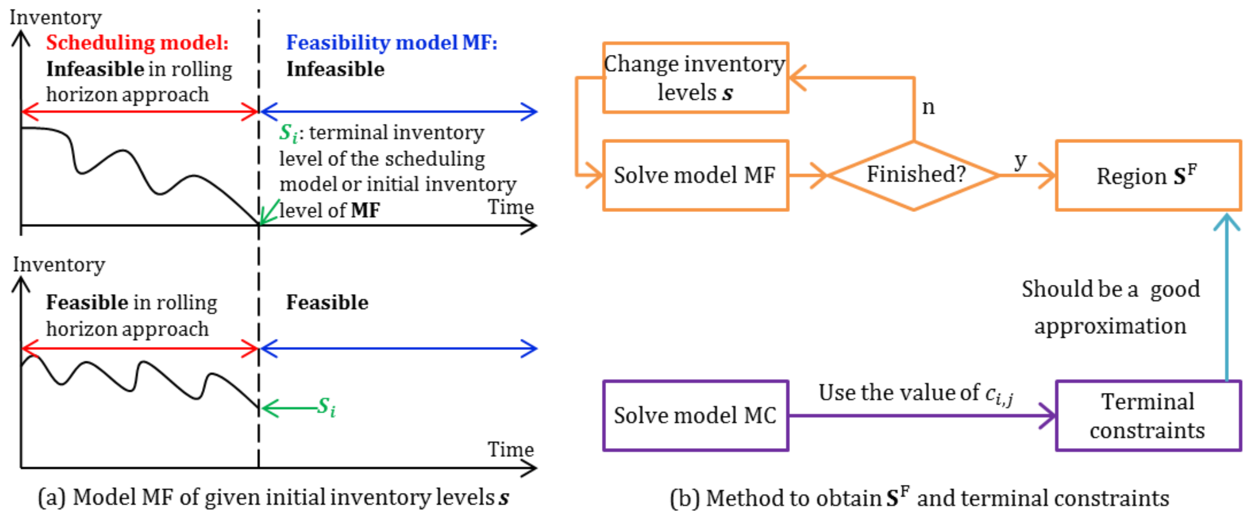


Figure 5.2. Model MF to check if $\boldsymbol{s} \in \mathbf{S}^{\mathrm{F}}$ and the overall approach.

Region $\mathbf{S}^{\mathrm{F}}$ might be non-convex, and therefore should be approximated by a set of linear terminal constraints when used in the scheduling model. Moreover, for a network with many materials, the iterative process to obtain $\mathbf{S}^{\mathrm{F}}$ might be intractable. Thus, we obtain region $\mathbf{S}^{\mathrm{F}}$ through iterations for simple networks, and find the terminal constraints to approximate $\mathbf{S}^{\mathrm{F}}$. Once we understand the logic behind these constraints (for simple networks), we can write them generally for networks with high dimensionality of materials. In the rest of the text, we use $\mathbf{S}^{(\mathrm{X})}$ to denote the feasible region subject to terminal constraints numbered (5.X). Good terminal constraints should lead to a close approximation of region $\mathbf{S}^{\mathrm{F}}$ in two aspects: (1) $\mathbf{S}^{(\mathrm{X})}$ should be a subset of $\mathbf{S}^{\mathrm{F}}$; and (2) $\mathbf{S}^{\mathrm{F}} \setminus \mathbf{S}^{(\mathrm{X})}$ should be as small as possible.

To write our terminal constraints, we need to find out some “hidden” parameters, which are revealed by solving a campaign model MC (given in §5.2.3). The value of variable c_{ij} in MC, denoting how many times task *i* should be executed in machine *j*, is to be used as a parameter in the terminal constraints (purple blocks in Figure 5.2(b)).

We propose terminal constraints for different network structures, from the simplest to the most general (Figure 5.3). The single-stage single-product problem is trivial, since it suffices to require that the terminal inventory is greater than the demand during the first batch of production; others will be studied in the following sections.

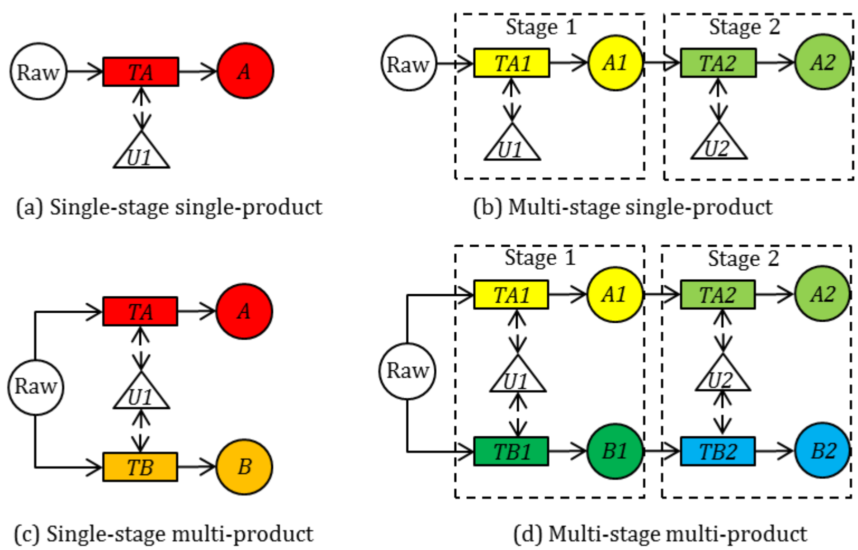


Figure 5.3. Different network structures.

5.2.2. Feasibility Model (MF)

As mentioned in §5.2.1, model MF is solved repeatedly with different values of $\boldsymbol{s}$ to obtain $\mathbf{S}^{\mathrm{F}}$. If $\boldsymbol{s} = [S_{m1}, S_{m2}, \dots, S_{m|\mathbf{M}|}]^{\mathrm{T}} \in \mathbf{S}^{\mathrm{F}}$, then $\left\{\boldsymbol{s}' = [S'_{m1}, S'_{m2}, \dots, S'_{m|\mathbf{M}|}]^{\mathrm{T}} \middle| S'_m \geq S_m, \forall m\right\} \subseteq \mathbf{S}^{\mathrm{F}}$; therefore, we do not need to check every point $\boldsymbol{s}$. The algorithm to obtain $\mathbf{S}^{\mathrm{F}}$ is shown in Appendix.

Model MF involves task-machine assignment and timing decisions. The variables include:

(a) $W_{ijt} \in \{0,1\}$: =1 if and only if task *i* starts in machine *j* at time point *t*;

(b) $L_{mt} \in \mathbb{R}^{+}$: inventory level of material *m* during time period *t*.

Model MF is as follows,

Minimize $$\sum_{i,j\in\mathbf{J}_i,t}\alpha_{ij}W_{ijt}+\sum_{m,t}\pi_m L_{mt} \tag{5.1a}$$

Subject to $$L_{m,1}=S_m-\sum_{i\in\mathbf{I}_m^-,j\in\mathbf{J}_i}\beta_{ij}W_{ij,0}-\delta_m \quad \forall m \tag{5.1b}$$

$$L_{m,t+1}=L_{mt}+\sum_{i\in\mathbf{I}_m^+,j\in\mathbf{J}_i}\beta_{ij}W_{ij,t-\tau_{ij}}-\sum_{i\in\mathbf{I}_m^-,j\in\mathbf{J}_i}\beta_{ij}W_{ijt}-\delta_m \quad \forall m,t>0 \tag{5.1c}$$

$$\sum_{i\in\mathbf{I}_j,t-\tau_{ij}+1\le t'\le t}W_{ijt'}\le 1 \quad \forall j,t \tag{5.1d}$$

The objective function (5.1a) is to minimize production cost and inventory holding cost. Material balance is expressed in constraints (5.1b) and (5.1c). Constraints (5.1d) enforce that only one task can be processed in a certain machine at each time. Note that when checking the feasibility of model MF, horizon length should be long enough.

Proposition 5.1 below shows that if model MF is feasible with the given $\boldsymbol{s}$, then a scheduling problem whose terminal inventory levels are equal to $\boldsymbol{s}$ will be recursively feasible (i.e., $\boldsymbol{s}\in\mathbf{S}^{\mathrm{F}}$). The proof is given in Appendix A.

Proposition 5.1*:* Let $L_{mt}(S1)$ and $W_{ijt}(S1)$ be the values from a feasible solution, $S1$, of model MF. If model MF has a feasible solution, $S2$, when using $S_m=L_{m,T+1}(S1)+\delta_m+\sum_{i\in\mathbf{I}_m^-,j\in\mathbf{J}_i}\beta_{ij}W_{ij,T}(S1)$, then for any $1<\sigma\le T+1$, using $S_m=L_{m,\sigma}(S1)+\delta_m+\sum_{i\in\mathbf{I}_m^-,j\in\mathbf{J}_i}\beta_{ij}W_{ij,\sigma-1}(S1)$, model MF also has a feasible solution, $S3$.

5.2.3. Campaign Model (MC)

Before writing the terminal constraints, we need to find how frequently each task is carried out in a "typical" scheduling solution. This is achieved by solving an auxiliary linear programming (LP) model MC. The variables include:

(a) $c_{ij}\in\mathbb{R}^+$: number of batches that task *i* is processed in machine *j*;

(b) $H\in\mathbb{R}^+$: campaign time.

The value of c_{ij} is to be used as a parameter when writing the terminal constraints. We introduce a new parameter used in this model: $\hat{\delta}_m$ denotes the "propagated" normalized demand. For product $m' \in \mathbf{M}^P$, $\hat{\delta}_{m'} = \delta_{m'}$; for an intermediate material m produced in stage k, $\hat{\delta}_m = \sum_{m':m=m(m',k)} \delta_{m'}$. Model MC is as follows,

Minimize
$$\sum_{m,i\in\mathbf{I}_m^+,j\in\mathbf{J}_i} \frac{\beta_{ij}}{\hat{\delta}_m} c_{ij} \tag{5.2a}$$

Subject to
$$H \geq \sum_{i\in\mathbf{I}_j} \tau_{ij} c_{ij} \quad \forall j \tag{5.2b}$$

$$\sum_{i\in\mathbf{I}_m^+,j\in\mathbf{J}_i} \beta_{ij} c_{ij} \geq \hat{\delta}_m H \quad \forall m \tag{5.2c}$$

$$\sum_{i\in\mathbf{I}_m^+,j\in\mathbf{J}_i} c_{ij} \geq 1 \quad \forall m \tag{5.2d}$$

The objective function (5.2a) minimizes the total production in a campaign (normalized to the demand of each material). In constraints (5.2b), the campaign time is required to be greater than the total production time for each machine. The production amount should be greater than the demand, as shown in constraints (5.2c). To avoid the trivial solution in which all variables are zero, constraints (5.2d) requires that each material is produced at least once. Because the values of c_{ij} appear linearly on both sides of the proposed terminal constraints (presented in §5.4 and §5.5), it is their relative ratios that are important, and therefore variables c_{ij} are defined to be continuous, rather than integer. Because we assumed that demand can be fulfilled by production, model MC is always feasible.

5.3. Multi-stage Single-product Problems

The problem addressed in this section is similar to the flow shop scheduling problem with only one type of product. To simplify the notation in this section, we drop index j, since there is only one machine in each stage. Also, each material can be represented by the stage in which it is produced, and each task can be represented by the stage it belongs to. Thus, we replace both indices *m* and *i* with k (using symbols of $\beta_k, \tau_k, W_{kt}, S_k, L_{kt}$), and use δ to denote the normalized demand of the product. For the problem in this section, solving model MC is not needed.

5.3.1. Proposed Terminal Constraints

Starting from a case of two stages (shown in Figure 5.3(b)), model MF is feasible if initial inventory levels S_k satisfy the following two constraints:

$$\frac{S_2}{\delta} \geq \tau_2$$

$$\frac{S_1 + S_2}{\delta} \geq \tau_1 + \tau_2 + \frac{\beta_2}{\delta}$$

We define the normalized inventory $\hat{S}_k = S_k/\delta$, denoting the number of periods for which the inventory itself can meet the demand. The first constraint requires that the initial normalized inventory of the product should be greater than the processing time of stage 2 so that the demand before the finishing of the first batch can be satisfied. In the second constraint, if we view stages 1 and 2 together as a "*pseudo-stage*", the left hand side (LHS) can be viewed as the "*propagated*" inventory (from stage 1 to stage 2), and the right hand side (RHS) as the "*propagated*" lead time, which is the summation of processing times plus the batch size of stage 2 divided by the demand. Note that the last term, β_2/δ, is added because model MF could be infeasible without it, as there might be a gap in the production between the two stages when β_1 is less than β_2 (shown in Figure 5.1(a) of the motivating example).

More generally, the terminal constraints can be written as follows,

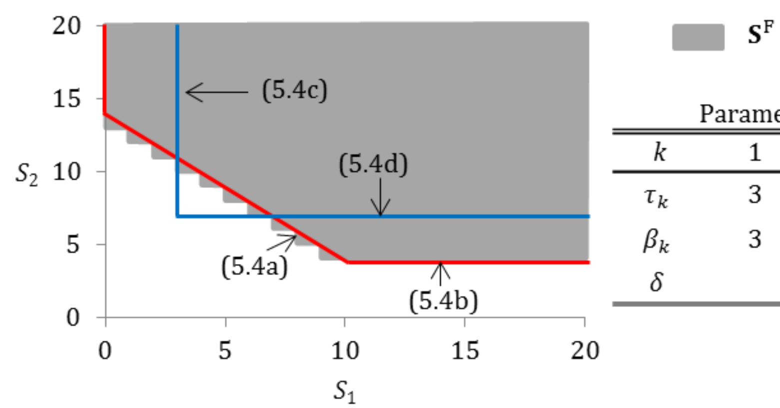
$$\sum_{k'=k}^{|\mathbf{K}|} \frac{S_{k'}}{\delta} \geq \sum_{k'=k}^{|\mathbf{K}|} \tau_{k'} + \sum_{k'=k+1}^{|\mathbf{K}|} \frac{\beta_{k'}}{\delta} \quad \forall k \tag{5.3}$$

There are $|\mathbf{K}|$ constraints included in (5.3). For the constraint written for stage k, the LHS is the propagated normalized inventory (from stage k to the final stage $|\mathbf{K}|$), while the RHS is the propagated lead time.

In Appendix B, we prove Proposition 5.2 below, which shows that if the initial inventory levels satisfy constraints (5.3), model MF is guaranteed to be feasible. Together will Proposition 5.1, we know the feasibly region subject to constraints (5.3) is a subset of $\mathbf{S}^{\mathrm{F}}$ ($\mathbf{S}^{(3)} \subseteq \mathbf{S}^{\mathrm{F}}$); i.e., constraints (5.3) lead to recursive feasibility for model MF (Corollary 5.3). The proposed terminal constraints are better than the traditional approach, because the feasible region subject to the traditional threshold constraints is not always a subset of $\mathbf{S}^{\mathrm{F}}$ (shown in §5.3.2).

Proposition 5.2*:* For multi-stage single-product problems, if the initial inventory levels S_k satisfy constraints (5.3), model MF is always feasible regardless of the horizon length.

Corollary 5.3*:* For multi-stage single-product problems, if the terminal inventory levels satisfy constraints (5.3), model MF is recursively feasible.



Parameters

k	1	2
τ_k	3	4
β_k	3	7
δ		1

Figure 5.4. Parameters, region $\mathbf{S}^{\mathrm{F}}$, proposed terminal constraints, and the traditional thresholds for the 2-stage example.

5.3.2. Examples

We consider a 2-stage example (parameters in Figure 5.4). Following constraints (5.3), the proposed terminal constraints are:

$$S_1 + S_2 \geq 14 \tag{5.4a}$$

$$S_2 \geq 4 \tag{5.4b}$$

Using the traditional threshold approach, the terminal inventory levels are constrained as follows,

$$S_1 \geq 3 \tag{5.4c}$$

$$S_2 \geq 7 \tag{5.4d}$$

Region $\mathbf{S}^{\mathrm{F}}$, obtained by repeatedly solving model MF with different initial inventory levels, is shown in Figure 5.4; we also show the proposed terminal constraints defined in constraints (5.4a), (5.4b), as well as the constraints based upon the traditional thresholds (5.4c) and (5.4d). The feasible region defined by (5.4a), (5.4b) (together with the non-negativity of S_1) is included in region $\mathbf{S}^{\mathrm{F}}$, and is a very close approximation of $\mathbf{S}^{\mathrm{F}}$. On the other hand, the feasible region subject to the traditional thresholds is not entirely in region $\mathbf{S}^{\mathrm{F}}$, which is the reason of the stockout shown in §5.1.1.

5.4. Single-stage Multi-product Problems

The problem addressed in this section is similar to the single-machine problem for discrete manufacturing. To simplify the notations in this section, each material can be represented by the task that produces it. Thus, we replace index *m* by index *i* (using notation of δ_i, S_i, L_{it}), and drop indices *j* and *k*. We propose two types of terminal constraints that lead to the same feasible region. The first type includes more constraints compared to the second, but requires no auxiliary variables.

5.4.1. Type 1 Terminal Constraints

Starting from a case of two products (shown in Figure 5.3(c)), model MF is feasible if initial inventory levels S_i satisfy the following three constraints:

$$\frac{S_A}{\delta_A} \geq \tau_A \tag{5.5a}$$

$$\frac{S_B}{\delta_B} \geq \tau_B \tag{5.5b}$$

$$\frac{c_A \tau_A S_A}{\delta_A} + \frac{c_B \tau_B S_B}{\delta_B} \geq (c_A \tau_A + c_B \tau_B)(\tau_A + \tau_B) \tag{5.5c}$$

where c_A and c_B are obtained by solving model MC.

Similarly as in §5.3.1, we define the normalized inventory $\hat{S}_i = S_i/\delta_i$. Constraints (5.5a) and (5.5b) requires that the normalized inventory should be greater than or equal to the processing time, so that the inventory is sufficient to last during the execution of the first batch. To interpret constraint (5.5c), we define $\rho_i = c_i \tau_i$ denoting the production time of a product for $i \in \{A, B\}$, and $\rho_{A+B} = c_A \tau_A + c_B \tau_B$ denoting the total production time of $A + B$. Thus, constraint (5.5c) can be rewritten as follows,

$$\frac{\rho_A}{\rho_{A+B}} \hat{S}_A + \frac{\rho_B}{\rho_{A+B}} \hat{S}_B \geq \tau_A + \tau_B \tag{5.5d}$$

If we view $A + B$ as a "*pseudo-product*", constraint (5.5d) can be interpreted as a generalization of (5.5a). The RHS is the processing time of the pseudo-product; while the LHS is the normalized inventory of the pseudo-product, which is a weighted summation of the inventory of products. The weight is the ratio of the production time of a product to the production time of all the products in the pseudo-product. If product i requires a longer production time, the inventory of i, $\hat{S}_i$, plays a more important role, and thus the weight is heavier.

More generally, the terminal constraints can be written as,

$$\sum_{i \in \mathbf{I}_p} \frac{c_i \tau_i S_i}{\delta_i} \geq \left(\sum_{i \in \mathbf{I}_p} c_i \tau_i \right) \cdot \left(\sum_{i \in \mathbf{I}_p} \tau_i \right), \forall p \in \mathbf{P}(\mathbf{I}) \tag{5.6}$$

where $\mathbf{P}(\mathbf{I})$ denotes the power set of $\mathbf{I}$ (i.e., the set of all subsets of $\mathbf{I}$) except the empty set, indexed by p; $\mathbf{I}_p$ denotes the elements that are included in the subset p. The interpretation of constraints

(5.6) follows the same logic as we discussed for constraints (5.5d). Because constraints (5.6) are written for each subset of **I** except the empty set, the total number of constraints is $2^{|\mathbf{I}|} - 1$.

In Appendix C, we prove Proposition 5.4, which shows that if the initial inventory levels satisfy constraints (5.6), model MF is guaranteed to be feasible. Together with Proposition 5.1, we know that the constraints lead to recursive feasibility for model MF (Corollary 5.5).

Proposition 5.4*:* For single-stage multi-product problems, if the initial inventory levels S_i satisfy constraints (5.6), model MF is always feasible regardless of the horizon length.

Corollary 5.5*:* For single-stage multi-product problems, if the terminal inventory levels satisfy constraints (5.6), model MF is recursively feasible.

5.4.2. Type 2 Terminal Constraints

Starting from a case of two products again, model MF is feasible if inventory levels S_i satisfy the following three constraints:

$$\frac{S_A}{\delta_A} \geq \tau_A + \tau_{\mathrm{B}} \frac{c_B}{c_A} U_{A,B} \tag{5.7a}$$

$$\frac{S_B}{\delta_B} \geq \tau_A\left(\frac{c_A}{c_B} + 1 - U_{A,B}\right) + \tau_B \tag{5.7b}$$

$$0 \leq U_{A,B} \leq \frac{c_A}{c_B} + 1 \tag{5.7c}$$

where an auxiliary continuous variable $U_{A,\mathrm{B}}$ is introduced. It can be shown that constraints (5.7a)-(5.7c) lead to the same feasible region as constraints (5.5a)-(5.5c), in terms of inventory levels S_i.

For general cases, we introduce auxiliary continuous variables $U_{ii'}$ for each *i* and *i'* such that *i'*>*i* (in terms of the orders in the set). The type 2 constraints are as follows,

$$\frac{S_i}{\delta_i} \geq \tau_i + \sum_{i'<i} \tau_{i'} \left(\frac{c_{i'}}{c_i} + 1 - U_{i'i}\right) + \sum_{i'>i} \tau_{i'} \frac{c_{i'}}{c_i} U_{ii'} \quad \forall i \tag{5.8a}$$

$$0 \leq U_{ii'} \leq \frac{c_i}{c_{i'}} + 1 \quad \forall i, i' > i \tag{5.8b}$$

which requires $|\mathbf{I}| \cdot (|\mathbf{I}| - 1)/2$ additional $U_{ii'}$ variables, but includes less constraints, $|\mathbf{I}| \cdot (|\mathbf{I}| + 1)/2$, compared to the first type.

In Appendix D, we prove Proposition 5.6, which shows that the terminal constraints proposed in §5.4.1 and §5.4.2 lead to the same feasible region of inventory levels. Thus, we can derive Corollary 5.7 based on Corollary 5.5 and Proposition 5.6 to show that type 2 constraints also lead to recursive feasibility for model MF.

***Proposition 5.6**:* The projection of feasible region defined by constraints (5.8a) and (5.8b) on the subspace of $\boldsymbol{s} = [S_1, S_2, \dots, S_{|\mathbf{I}|}]^{\mathrm{T}}$ is the same as the feasible region defined by constraints (5.6).

***Corollary 5.7**:* For single-stage multi-product problems, if the terminal inventory levels satisfy constraints (5.8a) and (5.8b), model MF is recursively feasible.

For the problems with many tasks, constraints (5.8a) and (5.8b) may perform better than constraints (5.6), because the number of constraints (5.6) grows exponentially with the number of tasks. Nevertheless, we will focus on the type 1 constraints in the rest of the chapter for the sake of brevity.

5.4.3. Examples

We consider a 2-product example (parameters in Figure 5.5). Solving model MC, we obtain $c_A = 2, c_B = 1$. The proposed terminal constraints (of type 1) are:

$$S_A \geq 4 \tag{5.9a}$$

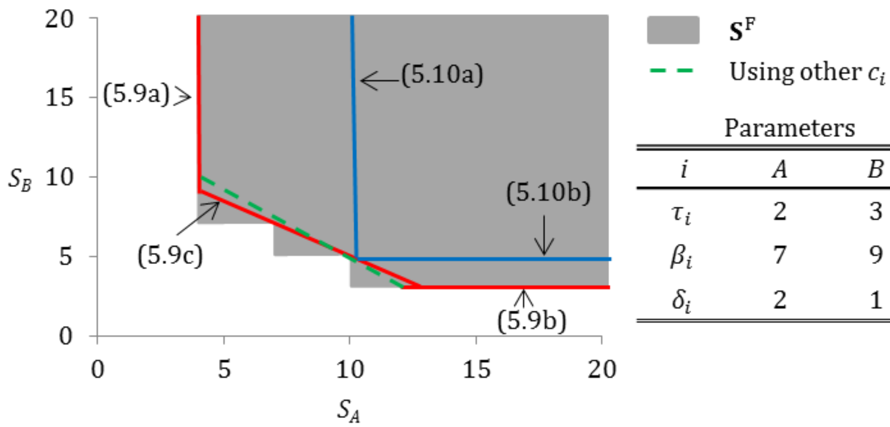
$$S_B \geq 3 \tag{5.9b}$$

$$2S_A + 3S_B \geq 35 \tag{5.9c}$$

Using the traditional threshold approach, the terminal inventory levels are constrained as follows,

$$S_A \geq 10 \tag{5.10a}$$

$$S_B \geq 5 \tag{5.10b}$$



Parameters

i	A	B
τ_i	2	3
β_i	7	9
δ_i	2	1

Figure 5.5. Parameters, region $\mathbf{S}^{\mathrm{F}}$, proposed terminal constraints, and the traditional thresholds for the 2-product example.

Region $\mathbf{S}^{\mathrm{F}}$, obtained by repeatedly solving MF with different initial inventory levels, is shown in Figure 5.5; we also show the proposed terminal constraints defined in constraints (5.9a)-(5.9c), the traditional constraints (5.10a)-(5.10b), as well as the terminal constraints using other values of c_i ($c_A = 2.571, c_B = 1$, obtained by solving a revised MC with a different objective function), which are feasible but not optimal for model MC. In this figure, we see that $\mathbf{S}^{(10a)(10b)} \subseteq \mathbf{S}^{(9a)(9b)(9c)} \subseteq \mathbf{S}^{\mathrm{F}}$; i.e., both the proposed terminal constraints and the traditional approach can lead to recursive feasibility, but the former has a larger feasible region than the latter. We also note that any c_i that is feasible for MC can be used to generate the terminal constraints, and neither the optimal c_i from MC nor the other feasible c_i leads to a better approximation.

Second, we consider an example with three products A, B, C, with $\tau_i = 2, \beta_i = 12, \delta_i = 2$ for $i \in \{A, B, C\}$. We obtain $c_A = c_B = c_C = 1$. There are seven terminal constraints in (5.6), written for subsets $\{A\}$, $\{B\}$, $\{C\}$, $\{A,B\}$, $\{A,C\}$, $\{B,C\}$, $\{A,B,C\}$. Using these terminal constraints, we approximate the non-convex region $\mathbf{S}^{\mathrm{F}}$ by a convex region $\mathbf{S}^{(6)}$, and $\mathbf{S}^{(6)} \subseteq \mathbf{S}^{\mathrm{F}}$ (Figure 5.6).

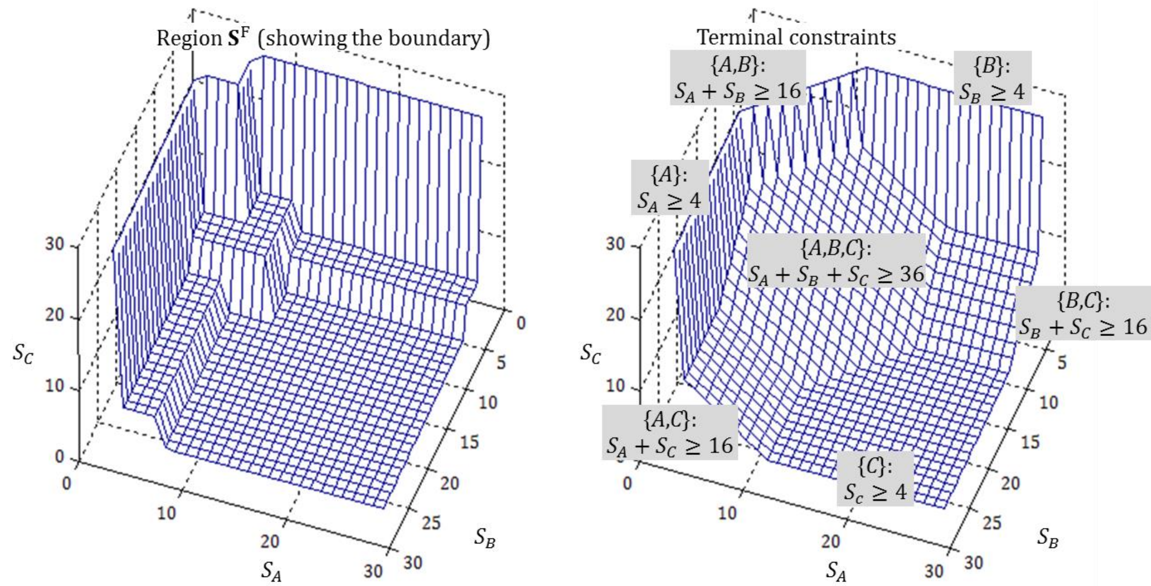


Figure 5.6. The boundary of region $\mathbf{S}^{\mathrm{F}}$ and the proposed terminal constraints for the 3-product example.

5.5. Multi-stage Multi-product Problems

The problem addressed in this section is similar to the flow shop scheduling problem. We drop index *j* once again. Both the multi-stage single-product problems and the single-stage multi-product problems are special cases of the multi-stage multi-product problems. Thus, the constraints we will present in this section can be viewed as a generalization of the constraints proposed in §5.3 and §5.4.

5.5.1. Proposed Terminal Constraints

Starting from 2-stage 2-product problems (shown in Figure 5.3(d)), we propose to use the following six constraints to constrain inventory levels S_m:

$$\frac{S_{A2}}{\delta_{A2}} \geq \tau_{TA2} \tag{5.11a}$$

$$\frac{S_{A1} + S_{A2}}{\delta_{A2}} \geq \tau_{TA1} + \tau_{TA2} + \frac{\beta_{TA2}}{\delta_{A2}} \tag{5.11b}$$

$$\frac{S_{B2}}{\delta_{B2}} \geq \tau_{TB2} \tag{5.11c}$$

$$\frac{S_{B1}+S_{B2}}{\delta_{B2}} \geq \tau_{TB1}+\tau_{TB2}+\frac{\beta_{TB2}}{\delta_{B2}} \tag{5.11d}$$

$$\frac{c_{TA2}\tau_{TA2}S_{A2}}{\delta_{A2}}+\frac{c_{TB2}\tau_{TB2}S_{B2}}{\delta_{B2}} \geq (c_{TA2}\tau_{TA2}+c_{TB2}\tau_{TB2})(\tau_{TA2}+\tau_{TB2}) \tag{5.11e}$$

$$\begin{aligned} &c_{TA1}\tau_{TA1}\frac{S_{A1}+S_{A2}}{\delta_{A2}}+c_{TB1}\tau_{TB1}\frac{S_{B1}+S_{B2}}{\delta_{B2}} \\ &\qquad \geq (c_{TA1}\tau_{TA1}+c_{TB1}\tau_{TB1})\left(\tau_{TA1}+\tau_{TA2}+\frac{\beta_{TA2}}{\delta_{A2}}+\tau_{TB1}+\tau_{TB2}+\frac{\beta_{TB2}}{\delta_{B2}}\right)\end{aligned} \tag{5.11f}$$

The 2-stage 2-product network can be decomposed into (1) two 2-stage single-product networks and (2) two single-stage 2-product networks. For the 2-stage single-product networks, following constraints (5.3), we write constraints (5.11a), (5.11b) for the production of $A2$, and constraints (5.11c) and (5.11d) for the production of $B2$. For the single-stage 2-product networks, we write constraint (5.11e) directly following constraints (5.6) for stage 2; while for stage 1, we write constraint (5.11f) using the idea of propagation introduced in §5.3. Specifically, the propagated inventory is used on the LHS of constraint (5.11f), considering the inventory levels of intermediates and products; subsequently, the term in the second parenthesis on the RHS is modified to the propagated lead time.

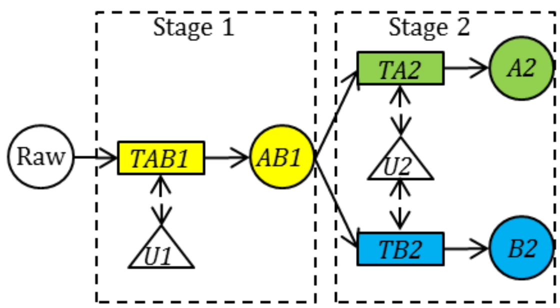
Generally, the terminal constraints are as follow,

$$\sum_{k'=k}^{|\mathbf{K}|}\frac{S_{m(m',k')}}{\delta_{m'}} \geq \sum_{k'=k}^{|\mathbf{K}|}\tau_{i(m',k')}+\sum_{k'=k+1}^{|\mathbf{K}|}\frac{\beta_{i(m',k')}}{\delta_{m'}} \quad \forall m' \in \mathbf{M}^P, k \tag{5.12a}$$

$$\begin{aligned} &\sum_{m'\in\mathbf{M}_p} c_{i(m',k)}\tau_{i(m',k)}\frac{\sum_{k'=k}^{|\mathbf{K}|}S_{m(m',k')}}{\delta_{m'}} \\ &\geq \left(\sum_{m'\in\mathbf{M}_p} c_{i(m',k)}\tau_{i(m',k)}\right)\cdot\sum_{m'\in\mathbf{M}_p}\left(\sum_{k'=k}^{|\mathbf{K}|}\tau_{i(m',k')}+\frac{\sum_{k'=k+1}^{|\mathbf{K}|}\beta_{i(m',k')}}{\delta_{m'}}\right) \\ &\qquad \forall p \in \mathbf{P}(\mathbf{M}^P) \text{ and } |\mathbf{M}_p| > 1, k \end{aligned} \tag{5.12b}$$

Constraints (5.12a) are written for the multi-stage single-product networks, where $i(m', k)$ denotes the task in stage k to produce product m', and $m(m', k)$ denotes the material produced in stage k for producing product m'. Constraints (5.12b) are written for the single-stage multi-product networks, where $\mathbf{P}(\mathbf{M}^P)$ denotes the power set of all products, $\mathbf{M}^P$, except the empty set, indexed by p; $\mathbf{M}_p$ denotes the products that are included in the subset p. Note that constraints (5.12b) are written for $|\mathbf{M}_p| > 1$, because the corresponding constraints of (5.12b) written for $|\mathbf{M}_p| = 1$ are already included in constraints (5.12a).

As shown in the following example, the proposed terminal constraints also apply to the network in which the same intermediate is used to produce different downstream materials.



Parameters

i	$TAB1$	$TA2$	$TB2$
τ_i	3	4	4
β_i	6	8	8
$\delta_m{}^*$		1	1

* Row of δ_m shows the demand of the product produced by the given task in the header.

Figure 5.7. Network and parameters for a 2-stage 2-product example with one intermediate.

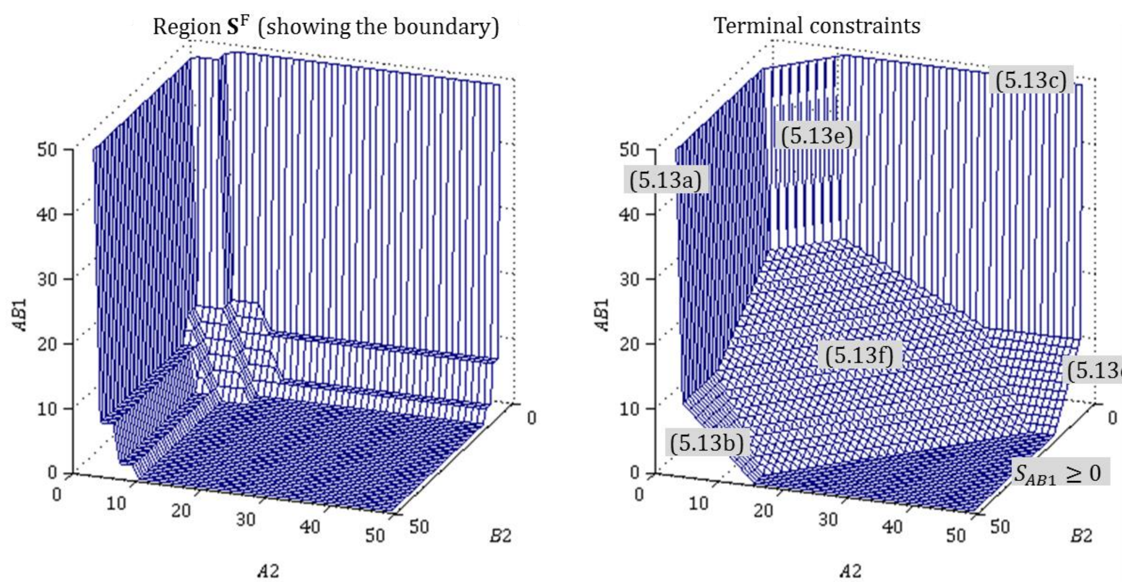


Figure 5.8. The boundary of region $\mathbf{S}^{\mathrm{F}}$ and the proposed terminal constraints for the 2-stage 2-product example (with one intermediate).

5.5.2. Examples

We consider a 2-stage 2-product example (Figure 5.7). Solving model MC, we obtain $c_{TAB1} = 1.33, c_{TA2} = c_{TB2} = 1$. Based on (5.12a) and (5.12b), the proposed terminal constraints are:

$$S_{A2} \geq 4 \tag{5.13a}$$

$$S_{AB1} + S_{A2} \geq 15 \tag{5.13b}$$

$$S_{B2} \geq 4 \tag{5.13c}$$

$$S_{AB1} + S_{B2} \geq 15 \tag{5.13d}$$

$$4S_{A2} + 4S_{B2} \geq 64 \tag{5.13e}$$

$$4(S_{AB1} + S_{A2}) + 4(S_{AB1} + S_{B2}) \geq 240 \tag{5.13f}$$

The inventory level of the intermediate, S_{AB1}, appears twice on the LHS of constraint (5.13f), because $AB1$ is used to produce both products. The six terminal constraints, together with the non-negativity of S_{AB1}, approximate the non-convex region $\mathbf{S}^{\mathrm{F}}$ by a convex region (Figure 5.8). The feasible region of the terminal constraints (with $S_{AB1} \geq 0$) is included in $\mathbf{S}^{\mathrm{F}}$.

5.6. Extension to Problems with Parallel Units

In §5.3-§5.5, we considered the problems with a single machine in each stage. In this section, we study problems with parallel machines in each stage. In §5.6.1 and §5.6.2, we follow the simplified notation used in §5.3 and §5.4 respectively.

5.6.1. Multi-stage Single-product Problems

5.6.1.1. Identical Units

If machines in the same stage are identical, we can still use the notation without index j. Following the same logic of Corollary 5.3, the following constraints can be shown to ensure recursive feasibility,

$$\sum_{k'=k}^{|\mathbf{K}|} \frac{S_{k'}}{\delta} \geq \sum_{k'=k}^{|\mathbf{K}|} \tau_{k'} + \sum_{k'=k+1}^{|\mathbf{K}|} \frac{|\mathbf{J}_{k'}| \cdot \beta_{k'}}{\delta} \quad \forall k \tag{5.14}$$

where the batch size is multiplied by the number of machines in the second summation on the RHS.

However, constraints (5.14) may not be a good approximation of $\mathbf{S}^{\mathrm{F}}$, and we want to use relaxed constraints to have a larger feasible region. By studying different examples, we observe that the original constraints (5.3) proposed in §5.3, which are relaxed constraints of (5.14), lead to a better approximation of $\mathbf{S}^{\mathrm{F}}$. Thus, we use constraints (5.3).

5.6.1.2. Non-identical Units

When machines in a certain stage are non-identical, we generalize constraints (5.3) as follows,

$$\sum_{k'=k}^{|\mathbf{K}|} \frac{S_{k'}}{\delta} \geq \sum_{k'=k}^{|\mathbf{K}|} \max_{j \in \mathbf{J}_{k'}} \tau_{k'j} + \sum_{k'=k+1}^{|\mathbf{K}|} \frac{\max_{j \in \mathbf{J}_{k'}} \beta_{k'j}}{\delta} \quad \forall k \tag{5.15}$$

in which the maximum processing time and the maximum batch size (over machines) are used in the first and second summations of the RHS respectively.

5.6.2. Single-stage Multi-product Problems

5.6.2.1. Identical Units

If all machines are identical, there is a solution of model MC with $c_{ij} = c_{ij'}$ for all *i*, *j*, *j'*. Thus, we can drop index *j* again. Constraints (5.6), proposed in §5.4, still lead to recursive feasibility, because when all the machines are synchronized to carry out the same task, the inventory profile will be the same as that in the single-machine case.

However, because machines are not required to be synchronized (i.e., we have more flexibility with multiple machines), constraints (5.6) are too conservative. To have a better approximation of region $\mathbf{S}^{\mathrm{F}}$, the terminal constraints are modified as follows,

$$\sum_{i\in\mathbf{I}_p}\frac{c_i\tau_i S_i}{\delta_i}\geq\sum_{i\in\mathbf{I}_p}c_i{\tau_i}^2+\mu\sum_{i\in\mathbf{I}_p,i'\in\mathbf{I}_p:i'\neq i}c_i\tau_i\tau_{i'}\quad\forall p\in\mathbf{P}(\mathbf{I})\tag{5.16}$$

where μ is a pre-defined parameter between 0 and 1. When $\mu = 1$, constraints (5.16) reduce to constraints (5.6). Based on our computational study, we use $\mu = 1/|\mathbf{J}|$, which leads to a good approximation of region $\mathbf{S}^{\mathrm{F}}$.

5.6.2.2. Non-identical Units

If machines are non-identical, the exact constraints for $|\mathbf{I}_p| = 1$ can be written as follows,

$$S_i\geq-\min_{0\leq l\leq\max_j\tau_{ij}-1}\left\{-\delta_i-l\delta_i+\sum_{j\in\mathbf{J}_i}\left\lfloor\frac{l}{\tau_{ij}}\right\rfloor\beta_{ij}\right\}\quad\forall i$$

The RHS represents the maximum backlog of product *i*, if its initial inventory is zero and its production is started in all machines at time 0. The other constraints ($|\mathbf{I}_p| > 1$) are harder to write, because processing times and batch sizes can vary among machines. Herein, we introduce the "average" parameters for each task *i*, (index *j* is again dropped,) as follows,

$$c_i=\sum_{j\in\mathbf{J}_i}c_{ij}\tag{5.17a}$$

$$\tau_i=\frac{1}{c_i}\sum_{j\in\mathbf{J}_i}c_{ij}\tau_{ij}\tag{5.17b}$$

With these average parameters, constraints (5.16) can be used.

5.6.3. Multi-stage Multi-product Problems

The terminal constraints of the most general network, with multiple stages, multiple products and multiple machines in each stage, can be obtained from the generalization of the constraints presented in §5.5.1, §5.6.1 and §5.6.2. We treat identical- and non-identical-machine problems in the same way. The following constraints are used,

$$\sum_{k'=k}^{|\mathbf{K}|} \frac{S_{m(m',k')}}{\delta_{m'}} \geq \sum_{k'=k}^{|\mathbf{K}|} \max_{j \in \mathbf{J}_{k'}} \tau_{i(m',k'),j} + \sum_{k'=k+1}^{|\mathbf{K}|} \frac{\max_{j \in \mathbf{J}_{k'}} \beta_{i(m',k'),j}}{\delta_{m'}} \quad \forall m' \in \mathbf{M}^P, k \tag{5.18a}$$

$$\sum_{m' \in \mathbf{M}_p} c_{i(m',k)} \tau_{i(m',k)} \frac{\sum_{k'=k}^{|\mathbf{K}|} S_{m(m',k')}}{\delta_{m'}} \geq$$

$$\sum_{m' \in \mathbf{M}_p} \left[c_{i(m',k)} \tau_{i(m',k)} \left(\sum_{k'=k}^{|\mathbf{K}|} \tau_{i(m',k')} + \frac{\sum_{k'=k+1}^{|\mathbf{K}|} \max_{j \in \mathbf{J}_{k'}} \beta_{i(m',k'),j}}{\delta_{m'}} \right) \right] \tag{5.18b}$$

$$+ \frac{1}{|\mathbf{J}_k|} \cdot \sum_{m' \in \mathbf{M}_p, m'' \in \mathbf{M}_p, m'' \neq m'} \left[c_{i(m',k)} \tau_{i(m',k)} \left(\sum_{k'=k}^{|\mathbf{K}|} \tau_{i(m'',k')} + \frac{\sum_{k'=k+1}^{|\mathbf{K}|} \max_{j \in \mathbf{J}_{k'}} \beta_{i(m'',k'),j}}{\delta_{m''}} \right) \right]$$

$$\forall p \subseteq \mathbf{P}(\mathbf{M}^P) \text{ and } |\mathbf{M}_p| > 1, k$$

In constraints (5.18b), the average parameter are calculated following the logic presented in §5.6.2.2, as follows,

$$c_{i(m',k)} = \sum_{j \in \mathbf{J}_{i(m',k)}} c_{i(m',k),j} \quad \forall m' \in \mathbf{M}^P, k \tag{5.18c}$$

$$\tau_{i(m',k)} = \frac{1}{c_{i(m',k)}} \cdot \sum_{j \in \mathbf{J}_{i(m',k)}} c_{i(m',k),j} \cdot \tau_{i(m',k),j} \quad \forall m' \in \mathbf{M}^P, k \tag{5.18d}$$

Terminal constraints for problems with different networks are summarized in Table 5.1.

Table 5.1. Terminal constraints for different problems.

	Single machine	Identical machines	Non-identical machines
Multi-stage single-product	(5.3)	(5.3)	(5.15)
Single-stage multi-product	(5.6)	(5.16)	(5.16)(5.17a)(5.17b)
Multi-stage multi-product	(5.12a)(5.12b)	(5.18a)(5.18b)	(5.18a)(5.18b)(5.18c)(5.18d)

5.7. Remarks

First, we comment on how to apply the terminal constraints when model MF is used as the scheduling model to solve an instance, rather than to study region $\mathbf{S}^{\mathrm{F}}$. The terminal constraints should constrain the terminal inventory level, rather than S_m, because S_m is a given parameter denoting the initial inventory level in the model. Moreover, even though we showed that the proposed terminal constraints ensure recursive feasibility for model MF, simply applying them for the inventory levels of the last time may cause stockout. This is because when solving model MF together with the terminal constraints, the model is modified (with terminal constraints added). Thus, simply applying the terminal constraints for the inventory levels of the last time cannot ensure recursive feasibility for the modified model. We note that the proposed terminal should be applied to the inventory levels of the last $\max_{i,j}\tau_{ij}$ times. By doing so, recursive feasibility is achieved for the modified model. For example, when applying terminal constraints (5.12a) in model MF, we should require the following constraints

$$\sum_{k'=k}^{|\mathbf{K}|}\frac{L_{m(m',k'),t}+\delta_{m(m',k')}+\sum_{i\in\mathbf{I}^{-}_{m(m',k')},j\in\mathbf{J}_i}\beta_{ij}W_{ij,t-1}}{\delta_{m'}}\geq\sum_{k'=k}^{|\mathbf{K}|}\tau_{i(m',k')}+\sum_{k'=k+1}^{|\mathbf{K}|}\frac{\beta_{i(m',k')}}{\delta_{m'}}$$

$$\forall m'\in\mathbf{M}^{P},k,t>T-\max_{i,j}\tau_{ij}$$

Note that the numerator of the LHS is not simply $L_{m(m',k'),t}$, because the numerator should represent the inventory level at time point t-1 (before the activity of consuming the product or the intermediate), which is the inventory level of period t, plus the normalized demand, $\delta_{m(m',k')}$, plus the consumption by the tasks in the following stage, $\sum_{i\in\mathbf{I}^{-}_{m(m',k')},j\in\mathbf{J}_i}\beta_{ij}W_{ij,t-1}$.

Second, when uncertainty is considered, buffer terms should be added. The main sources of uncertainty include the processing time and the demand. Based on safety stock literature (Eppen and Martin, 1988), if the mean and variance of the processing time and the normalized demand are denoted by $\bar{\tau}_{ij}$, $\bar{\delta}_m$, $\sigma^2(\tau_{ij})$ and $\sigma^2(\delta_m)$ respectively, then we define a buffer term B_m for every S_m in the terminal constraints:

$$B_m=\phi\sqrt{\sigma^2(\delta_m)\max_{i\in\mathbf{I}^{+}_m,j\in\mathbf{J}_i}\bar{\tau}_{ij}+(\bar{\delta}_m)^2\max_{i\in\mathbf{I}^{+}_m,j\in\mathbf{J}_i}\sigma^2(\tau_{ij})}\tag{5.19}$$

in which ϕ is the inverse distribution function of a standard normal distribution based on a specified service level. For example, after considering this buffer term, constraints (5.12a) becomes

$$\sum_{k'=k}^{|\mathbf{K}|} \frac{S_{m(m',k')} - B_{m(m',k')}}{\delta_{m'}} \geq \sum_{k'=k}^{|\mathbf{K}|} \tau_{i(m',k')} + \sum_{k'=k+1}^{|\mathbf{K}|} \frac{\beta_{i(m',k')}}{\delta_{m'}} \quad \forall m' \in \mathbf{M}^N, k$$

Third, when the demand is not constant but periodic, a simple change can be made. We assume the demand pattern repeats every $\gamma_{m'}$ periods for product m', and the average demand per period is $\delta_{m'}$. Then, we can apply the proposed constraints by requiring that the inventory levels of the last $\max(\max_{i,j} \tau_{ij}, \max_{m' \in \mathbf{M}^P} \gamma_{m'})$ times satisfy the terminal constraints.

5.8. Computational Results

In this section, we carry out simulations to study how the terminal constraints perform in online scheduling. We use model MF as the scheduling model. In order to continue the online scheduling when stockout happens, slack variables are introduced to allow negative inventory levels (backlogs), which are penalized in the objective function. We compare the solutions of three formulations:

(a) Model MF without any constraints on terminal inventory levels (referred as MF_{WO});

(b) Model MF with traditional threshold constraints (referred as MF_{TT}); and

(c) Model MF with the proposed terminal constraints (referred as MF_{TC}).

Applying the traditional threshold constraints or the proposed terminal constraints to the inventory levels of the last time would fail to prevent stockout. Thus, for MF_{TT} and MF_{TC}, we apply those two types of constraints to the inventory levels of the last $\max_{i,j} \tau_{ij}$ times, as discussed in §5.7.

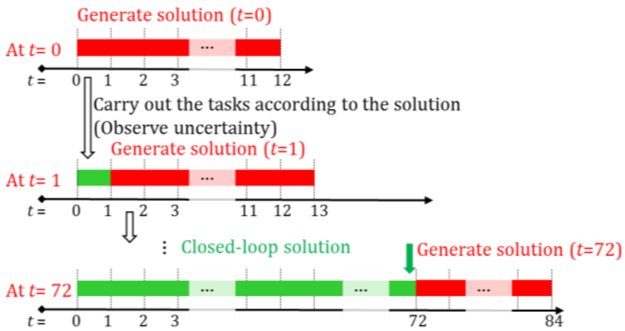


Figure 5.9. Online scheduling procedure.

The online scheduling procedure is shown in Figure 5.9. After obtaining a solution through optimization, we roll the horizon forward by one period and solve the scheduling model of the new horizon. If there is uncertainty, we observe its realization and update before rolling the horizon. After 72 iterations, we obtain the closed-loop solution from time 0 to time 72. After obtaining each closed-loop solution, we use two solution quality indicators:

(a) Stockout percentage (SP), which is the number of periods with negative inventory levels divided by the simulation horizon (72); and

(b) Average inventory levels (AIL), which is the average inventory levels (considering the summation of all materials in the network) over the periods.

We compare the three formulations based on SP and AIL; on condition that SP remains very low, the closed-loop solution is better if AIL has a lower value.

5.8.1. Deterministic Problem

We first consider deterministic problems, using three instances representing the three types of networks we discussed. Based on the closed-loop solution, the values of SP and AIL can be calculated (Table 5.2). When using MF_{WO}, the values of SP are large, and they decrease to zero by using MF_{TT} and MF_{TC}; this shows that constraints on terminal inventory levels are needed to prevent stockout. When using MF_{TC}, the values of AIL are smaller compared to those when using

MF_{TT}; this shows that the proposed terminal inventory levels can lead to lower inventory levels compared to the traditional approach.

5.8.2. Problems with Uncertainty

To further compare the solutions of the three formulations, we consider problems with uncertainty: the demand in each period is subject to a normal distribution, $\mathcal{N}(\delta_m, (0.3\delta_m)^2)$. Due to the uncertainty, inventory buffers were added in the model. First, L_{mt} was required to be greater than the buffer B_m defined in equation (5.19), which is $1.6 \cdot 0.3 \cdot \delta_m \cdot \sqrt{\max_{i \in \mathbf{I}_m^+, j \in \mathbf{J}_i} \tau_{ij}}$, for the uncertainty we consider; 1.6 is the value of ϕ used in equation (5.19) at a service level of 95%. Second, in the proposed terminal constraints and the traditional threshold constraints, the same buffer B_m was added.

Table 5.2. Values of SP (%) and AIL for the deterministic problems.

SP/AIL	Single-stage multi-product	Multi-stage single-product	Multi-stage multi-product
MF_{WO}	6.94/11.44	31.94/4.18	2.88/24.93
MF_{TT}	0.00/39.80	0.00/17.13	0.00/51.80
MF_{TC}	0.00/20.36	0.00/12.00	0.00/43.48

Table 5.3. The sample mean of SP (%) for the single-stage multi-product problem.

Instance	1	2	3	4	5	6	7	8
MF_{WO}	1.29	3.07	11.22	7.65	1.50	1.54	4.47	1.14
MF_{TT}	0.54	0.71	0.00	0.36	0.43	0.51	0.07	0.64
MF_{TC}	0.36	0.69	0.24	0.53	0.75	1.17	0.19	1.01

Table 5.4. The sample mean of SP (%) for the multi-stage single-product problem.

Instance	9	10	11	12	13	14	15	16
MF_{WO}	42.44	19.07	29.85	34.46	6.01	7.28	8.44	0.00
MF_{TT}	42.51	0.01	0.00	61.39	38.65	38.65	39.19	23.94
MF_{TC}	0.03	0.89	1.03	0.00	0.00	0.00	0.00	0.00

Table 5.5. The sample mean of SP (%) for the multi-stage multi-product problem.

Instance	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
MF_{WO}	43.53	32.81	10.72	9.04	5.01	25.58	36.51	7.13	6.26	2.49	1.58	1.90	1.03	1.42	1.99	2.72
MF_{TT}	37.85	0.00	0.00	13.19	0.00	0.00	0.19	0.00	29.96	5.86	7.93	9.29	4.43	8.22	9.38	0.82
MF_{TC}	0.13	0.01	0.51	0.04	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.15	0.00	0.04	0.03	0.03

We study 32 instances, which can be categorized into three types:

(a) Single-stage multi-product problem (instances 1-8);

(b) Multi-stage single-product problem (instances 9-16);

(c) Multi-stage multi-product problem (instances 17-32).

We obtained 100 samples for each instance; after obtaining the closed-loop solutions, we calculate the mean values of SP and AIL over the 100 samples. The sample means of SP are shown in Tables 5.3-5.5; while Figure 5.10 shows the increase (in percentage) in the sample mean of AIL using MF_{TT} compared to that using MF_{TC}, i.e., $AIL_{TT-TC} = (AIL(MF_{TT}) - AIL(MF_{TC}))/AIL(MF_{TC}) \cdot 100\%$. From the table and the figure, we can make the following observations:

(a) When no constraints on terminal inventory levels are applied, 26 out 32 instances have sample mean of SP greater than 1.5%, which re-confirm the necessity of constraining the terminal inventory levels;

(b) Taking SP<1.5% as the criterion of the effective prevention from stockout, the traditional threshold approach can prevent stockout for single-stage problems (Table 5.3); however, for 15 instances of other problems (Tables 5.4-5.5), it cannot prevent stockout;

(c) The proposed terminal constraints can prevent stockout for all instances;

(d) For the instances that both MF_{TC} and MF_{TT} can prevent stockout, MF_{TC} leads to lower inventory levels.

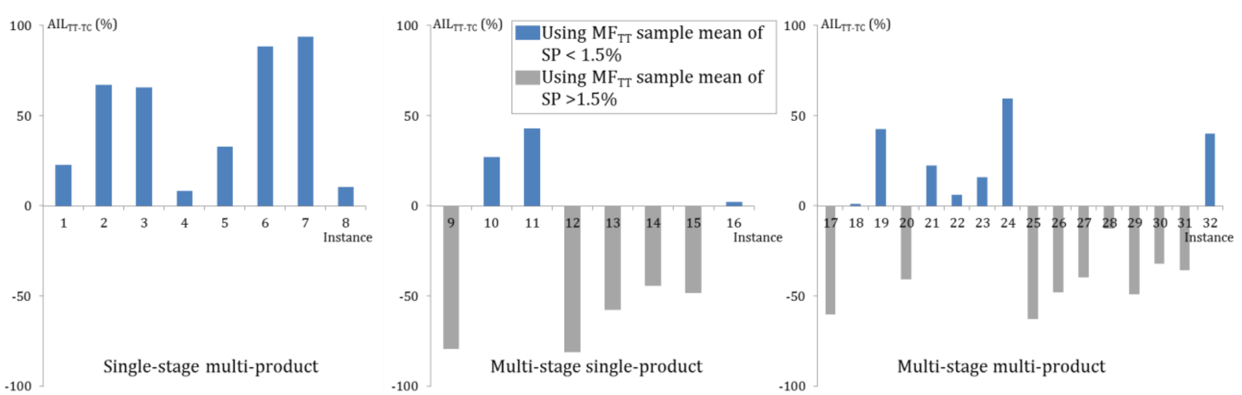


Figure 5.10. Increase of the sample mean of AIL using MF_{TT} compared to that using MF_{TC}.

5.9. Conclusions

We proposed novel terminal constraints for the production scheduling problems of different network structures, including multi-stage single-product networks, single-stage multi-product networks, and multi-stage multi-product networks. The proposed constraints consider the relationship of inventory levels of different materials. This is advantageous compared to the traditional threshold approach, which constrains the inventory levels independently for each material. Theoretically, we prove that for two types of networks, if the terminal inventory levels satisfy the proposed terminal constraints, the scheduling problem will be recursively feasible. By studying different problems with and without uncertainty, we show that the proposed terminal constraints can effectively prevent stockout, and achieve substantial savings on inventory holding cost by lowering the inventory levels, compared to the traditional approach. We also developed an approach to obtain the region of feasible terminal inventory levels, through iterations of solving a feasibility scheduling model. This approach can be generalized; i.e., one can obtain the region of feasible terminal inventory levels for other scheduling problems by iteratively solving the corresponding scheduling model.

5.10. Notation

Indices/Sets

$i \in \mathbf{I}$	tasks
$j \in \mathbf{J}$	machines
$k \in \mathbf{K}$	stages
$m \in \mathbf{M}$	materials
$\mathbf{M}^P \subseteq \mathbf{M}$	products
$p \in \mathbf{P}(\mathbf{M}^P)$	power set of the products (except the empty set)
$t \in \mathbf{T}$	time periods or points

Subsets

$\mathbf{I}_j$	tasks that can be carried out in machine *j*
$\mathbf{I}_m^+/\mathbf{I}_m^-$	tasks producing /consuming material *m*
$\mathbf{I}_p$	products (tasks) included in subset *p* (for single-stage problems)
$\mathbf{J}_i/\mathbf{J}_k$	machines that can carry out task *i* /tasks in stage *k*
$\mathbf{M}_p$	products included in subset *p*

Mappings

$i(m', k) \in \mathbf{I}$	task in stage *k* to produce product *m'*
$m(m', k) \in \mathbf{M}$	material produced in stage *k*, which is used to produce product *m'*

Parameters

α_{ij}	production cost of task *i* in machine *j*
β_{ij}	batch size of task *i* in machine *j*
$\delta_m/\hat{\delta}_m$	normalized/propagated demand of *m*
π_m	inventory holding cost of material *m* for one period
τ_{ij}	processing time of task *i* in machine *j*

Variables in model MF

$W_{ijt} \in \{0,1\}$	=1 if and only if task *i* starts in machine *j* at time point *t*
$L_{mt} \in \mathbb{R}^+$	inventory level of material *m* during time period *t*

Variables in model MC

$c_{ij} \in \mathbb{R}^+$	number of batches that task *i* is processed in machine *j*
$H \in \mathbb{R}^+$	campaign time.

Other notation

B_m	buffer term for uncertainty
S_m	terminal inventory level of material *m* (or the given initial inventory level when model MF is solved to obtain $\mathbf{S}^{\mathrm{F}}$)
$\boldsymbol{s}$	vector $[S_{m1}, S_{m2}, \dots, S_{m|\mathbf{M}|}]^{\mathrm{T}}$

$\mathbf{S}^{\mathrm{F}} \subseteq \mathbb{R}^{\lvert\mathbf{M}\rvert}$	region of feasible terminal inventory levels
$\mathbf{S}^{(\mathrm{X})} \subseteq \mathbb{R}^{\lvert\mathbf{M}\rvert}$	feasible region subject to constraints (5.X)
$U_{ii'}$	auxiliary continuous variable used in §5.4.2

Chapter 6

Discrete-time Formulations in Scheduling Problems with Changeover[6]

A wide range of mixed-integer programming (MIP) models (Wolsey, 1998) have been proposed in the literature to address manufacturing scheduling problems, and chemical production scheduling problems in particular (Méndez et al., 2006; Maravelias, 2012b; Harjunkoski et al., 2014). One of the differentiating attributes of the models is the *modeling of time*. In *discrete-time* models, the scheduling horizon η is divided into T periods of fixed length $\delta = \eta/T$, defining $T + 1$ time points (i.e., period t starts at time point $t - 1$ and ends at time point t). In continuous-time models, the horizon is divided into a known number of periods with variable length. Discrete-time models have several advantages over their continuous-time counterparts: they (1) are tighter (Sundaramoorthy and Maravelias, 2011a; Velez and Maravelias, 2013), (2) can easily handle intermediate release and due times, (3) can model holding and backlogging costs linearly, and (4) can be readily extended to handle events during the execution of a task. Furthermore, computational study showed that discrete-time models for problems in network production environments (i.e., environments where tasks produce and consume multiple materials and batches of materials can be mixed and split) can often be solved faster and find better solutions compared to continuous-time models (Sundaramoorthy and Maravelias, 2011a). However, because the size of discrete-time models grows at least linearly with the number of periods, the disadvantage of discrete-time models is the large number of binary variables and constraints, especially when sequence-dependent changeovers are considered. Such changeovers are common in the process industries (e.g., commodity, specialty, and fine chemicals; food and beverage manufacturing; pharmaceutical manufacturing; consumer goods), where cleaning-in-place, sterilization-in-place,

---

[6] This chapter is modified from Velez et al., 2017.

maintenance, material transfer, and unit setup activities need to be performed between different tasks.

If changeovers do not require resources and do not incur a cost, then changeover times can be enforced by simply allowing enough idle time between tasks (Kondili et al., 1993; Shah et al., 1993; Wolsey, 1997, Moniz et al., 2013). However, if resources are needed or costs need to be modeled, additional binary variables are necessary (Karmarkar and Schrage, 1985; Sahinidis and Grossmann, 1991; Kondili et al., 1993; Zentner et al., 1994). We study changeovers for processes that neither require resources nor incur a cost.

The chapter is structured as follows. In Section 6.1, we introduce the processes that we are interested in, describe their corresponding MIP models, and present three changeover formulations from the literature, and four changeover formulations proposed previously. In Section 6.2, we present a new formulation. Section 6.3 presents results regarding the relative tightness of the formulations. In Section 6.4, we present computational results. We use lowercase italic letters for indices, uppercase italic letters for variables, uppercase bold letters for sets, and lowercase Greek letters for parameters.

6.1. Background

We consider three variations on the single-stage environment with: (1) a single unit (machine), (2) parallel units, and (3) parallel units with unequal capacities. The horizon, η, is divided into T uniform intervals of length $\delta = \eta/T$, with $T + 1$ time points $t \in \{0,1, \dots, T\}$ occurring at $t\delta$. We use these three problems to study our new formulations because of their simplicity and because they represent three general classes of problems. We emphasize that the changeover constraints developed here can be readily used in *any* discrete-time model.

6.1.1. Single Unit

The single-unit problem consists of a set of tasks (jobs), $i \in \mathbf{I}$, with a fixed processing time, $\tau_i\delta$. We include a binary variable: $X_{it} = 1$ if and only if task i starts at time point t. We assume that each task must be run exactly once (constraints (6.1)) and only one task can run at a time (constraints (6.2)).

$$\sum_{t} X_{it} = 1 \quad \forall i \tag{6.1}$$

$$\sum_{i} \sum_{t'=t-\tau_i+1}^{t} X_{it'} \leq 1 \quad \forall t \tag{6.2}$$

We can minimize makespan, $MS \in \mathbb{R}^+$,

$$MS \geq \sum_{i} (t + \tau_i) X_{it} \quad \forall t \tag{6.3}$$

We can also minimize tardiness, $TRD \in \mathbb{R}^+$,

$$TRD = \sum_{i} TRD_i \tag{6.4}$$

where $TRD_i \in \mathbb{R}^+$ denotes the tardiness of task i. TRD_i is defined in constraints (6.5), where ϕ_i is the due time of task i:

$$TRD_i \geq \sum_{t=\phi_i-\tau_i+1}^{T} (t - \phi_i + \tau_i) X_{it} \quad \forall i \tag{6.5}$$

6.1.2. Parallel Units

When there are parallel units (machines), we introduce a new index, $j \in \mathbf{J}$, for units; the processing times, τ_{ij}, are unit-dependent. To generate a schedule we must assign each task to a unit, so the binary variables are indexed by j: $X_{ijt} = 1$ if and only if task i starts in unit j at time point t. We still assume that each task must be run exactly once (constraints (6.6), where $\mathbf{J}_i$ is the set of units that can process task i), and only one task can run at a time on a unit (constraints (6.7)).

$$\sum_{j \in \mathbf{J}_i, t} X_{ijt} = 1 \quad \forall i \tag{6.6}$$

$$\sum_{i} \sum_{t'=t-\tau_{ij}+1}^{t} X_{ijt'} \leq 1 \quad \forall j, t \tag{6.7}$$

Again, we consider makespan minimization,

$$MS \geq \sum_{i} (t + \tau_{ij}) X_{ijt} \quad \forall j, t \tag{6.8}$$

Since a task may run on different units and the cost may be different in each unit, we also consider cost, $CST \in \mathbb{R}^{+}$, minimization, where α_{ij} is the cost to run task *i* in unit *j*.

$$CST = \sum_{i,j,t} \alpha_{ij} X_{ijt} \tag{6.9}$$

For cost minimization, we consider the case where each task *i* has a hard due date (deadline), $\bar{\phi}_i$,

$$X_{ijt} = 0 \quad \forall i, j, t > \bar{\phi}_i - \tau_i \tag{6.10}$$

We also consider tardiness minimization, TRD, with constraints (6.4) and (6.11)

$$TRD_i \geq \sum_{j} \sum_{t=\phi_i-\tau_{ij}+1}^{T} (t - \phi_i + \tau_{ij}) X_{ijt} \quad \forall i \tag{6.11}$$

6.1.3. Parallel Units with Unequal Capacities

Each unit *j* has capacity β_j and (the output of) each task *i* has known demand, ξ_i. When units have unequal capacities, we cannot calculate how many times each task must run prior to optimization, so we replace constraints (6.6) with,

$$\sum_{j \in \mathbf{J}_i, t} \beta_j X_{ijt} \geq \xi_i \quad \forall i \tag{6.12}$$

Constraints (6.7) are also included to ensure that each unit processes at most one task at a time. Again, we consider *MS*, *CST*, and *TRD* minimization. The constraints for different problems and objective functions are summarized in Table 6.1.

Table 6.1. Constraints used in different production environments and objective functions.

Production environments	min *MS*	min *TRD*	min *CST*
Single unit	(6.1) (6.2) (6.3)	(6.1) (6.2) (6.4) (6.5)	
Parallel units	(6.6) (6.7) (6.8)	(6.4) (6.6) (6.7) (6.11)	(6.6) (6.7) (6.9) (6.10)
Parallel units with unequal capacities	(6.7) (6.8) (6.12)	(6.4) (6.7) (6.11) (6.12)	(6.7) (6.9) (6.10) (6.12)

6.1.4. Assumptions and Literature Formulations

The changeover time after task i finishes and before task i' starts on unit j is denoted by $\sigma_{ii'j}$ in terms of the number of periods. We make three assumptions.

Assumption 1. The changeover time between task i and task i' is less than both τ_{ij} and $\tau_{i'j}$ ($\sigma_{ii'j} < \min\{\tau_{ij}, \tau_{i'j}\}$).

Assumption 2. Changeover times satisfy the triangle inequality ($\sigma_{ii''j} < \sigma_{ii'j} + \tau_{i'j} + \sigma_{i'i''j}$).

Assumption 3. Tasks do not have a changeover with themselves ($\sigma_{iij} = 0$).

For the single-unit problem, one drops the index j in τ_{ij} and $\sigma_{ii'j}$, and the assumptions are then analogous to those for the parallel-units problem. The changeover constraints presented in the remaining of the chapter can be written for the single-unit problem, simply by replacing variables X_{ijt} by variables X_{it}. Also, when processing and/or changeover times are unit independent, all the constraints in the chapter still apply.

Three discrete-time formulations have appeared in the literature to enforce changeover times without additional binary variables.

Constraints (K). Kondili et al. (Kondili et al., 1993) used a big-M constraint:

$$\sum_{i' \neq i} \sum_{t'=t+\tau_{ij}}^{t+\tau_{ij}+\sigma_{ii'j}-1} X_{i'jt'} \leq M(1 - X_{ijt}) \quad \forall i, j, t \tag{6.13}$$

Although no value for M was suggested, one that is obviously large enough is

$$M_{ij} = \sum_{i' \neq i} \sum_{t'=t+\tau_{ij}}^{t+\tau_{ij}+\sigma_{ii'j}-1} 1 = \sum_{i' \neq i} \sigma_{ii'j} \tag{6.14}$$

While smaller values of M can be found, we observed that the value of M does not make a significant difference in the solution time.

Constraints (SH). Shah et al. (Shah et al., 1993) eliminated the big-M constraint by considering pair of tasks:

$$X_{ijt} + X_{i'jt'} \leq 1 \quad \forall i, i' \neq i, j, t, t - \tau_{i'j} - \sigma_{i'ij} < t' \leq t - \tau_{i'j} \tag{6.15}$$

Constraints (W). Finally, Wolsey (Wolsey, 1997) proposed the following constraints,

$$\sum_{t'=t-\tau_{ij}-\sigma_{ii'j}+1}^{t} X_{ijt'} + \sum_{t'=t-\tau_{i'j}-\sigma_{i'ij}+1}^{t} X_{i'jt'} \leq 1 \quad \forall i, i' \neq i, j, t \tag{6.16}$$

Constraints (6.16) include the binary variable for task i for more than τ_{ij} consecutive time points, which prevents task i from occurring back-to-back, so they are valid only when each task is restricted to run once. In the chapter, we also consider problems where multiple executions of the same task are allowed (in problems of parallel units with unequal capacities).

Three discrete-time formulations have appeared in the literature to enforce changeover times without additional binary variables.

Velez presented 4 changeover formulations (Velez, 2014), classified based on:

(a) The number of tasks for which each changeover constraint is written: a single task (S) or a pair of tasks (P);

(b) The number of time points for which each changeover constraint is written: all pairs of time points within an interval depending on the processing and changeover times (I), a subset of pairs of time points (II), or a single time point (III).

For instance, the three formulations presented previously can be classified accordingly: constraints (K) are type (S)/(III); constraints (SH) are type (P)/(I); constraints (W) are type (P)/(III). The 5 new formulations we present in §3.1-3.5 are named based on their classification: (SI), (SII), (SIIT) (a tighter version of (SII)), (SIII), and (P)=(PI)=(PII)= (PIII).

The following sets are introduced,

(a) $\mathbf{T}^P_{ijt} = \{t' | t - \tau_{ij} + 1 \leq t' \leq t\}$, referred to as the set of *processing time points* for *i, j, t*.

(b) $\mathbf{T}^C_{i'ijt} = \{t' | t - \tau_{i'j} - \sigma_{i'ij} + 1 \leq t' \leq t - \tau_{i'j}\}$, referred to as the set of *changeover time points* from *i'* to *i* for *j, t*.

Considering a given task *i*, time point *t*, and unit *j*, constraints (6.7) enforce $X_{i'jt'} = 0$, for each $i' \in \mathbf{I}$, $t' \in \mathbf{T}^P_{i'jt}$, if $X_{ijt} = 1$; while changeover constraints should enforce $X_{i'jt'} = 0$, for $i' \in \mathbf{I} \backslash \{i\}$, $t' \in \mathbf{T}^C_{i'ijt}$, if $X_{ijt} = 1$. Thus, variables $X_{i'jt'}$ with $t' \in \mathbf{T}^C_{i'ijt}$ should be included in at least one changeover constraint. Including $X_{i'jt'}$ with $t' \in \mathbf{T}^P_{i'jt}$ may tighten the changeover constraints, but it is not necessary.

Constraints (SI). Constraints (SI) are written for a single task and unit and for pairs of time points, as follows:

$$X_{ijt} + \sum_{i' \neq i} \sum_{t''=\max\{t', t-\sigma_{i'ij}\}-\tau_{i'j}+1}^{\min\{t', t-\tau_{i'j}\}} X_{i'jt''} \leq 1 \tag{6.17}$$

$$\forall i, j, t, t - \max\left\{\min_{i' \neq i} \tau_{i'j}, \max_{i' \neq i} \sigma_{i'ij}\right\} \leq t' \leq t - \min_{i' \neq i} \tau_{i'j}$$

Constraints (SII). We use integer parameter ν_{ijn}, where *n* indexes constraints (SII) and $\mathbf{N}_{ij}$ is the set of indices *n* for which ν_{ijn} is defined:

$$X_{ijt} + \sum_{i' \neq i} \sum_{t''=\max\left\{t-\nu_{ijn}, t-\sigma_{i'ij}\right\}-\tau_{i'j}+1}^{\min\left\{t-\nu_{ijn}, t-\tau_{i'j}\right\}} X_{i'jt''} \leq 1 \quad \forall i, j, t, n \in \mathbf{N}_{ij} \tag{6.18}$$

Parameter ν_{ijn} is defined as follows

$$\nu_{ij1} = \min_{i' \neq i} \tau_{i'j} \tag{6.19}$$

$$\nu_{ij,n+1} = \nu_{ijn} + \min_{i':\nu_{ij,n} < \sigma_{i'ij}} \tau_{i'j} \tag{6.20}$$

The largest index *n* for an (i, j) pair satisfies that ν_{ijn} is greater than or equal to the changeover from any task to i, i.e., $\nu_{ij,|\mathbf{N}_{ij}|} \geq \max_{i' \neq i} \sigma_{i'ij}$.

Constraints (SIII). To write a constraint for a single time point, we sum constraints (SII) over *n* to obtain a big-M constraint with $|\mathbf{N}_{ij}|$ as the big-M parameter,

$$\sum_{i' \neq i} \sum_{t'=t-\tau_{i'j}-\sigma_{i'ij}+1}^{t-\tau_{i'j}} X_{i'jt'} \leq |\mathbf{N}_{ij}|(1 - X_{ijt}) \quad \forall i, j, t \tag{6.21}$$

Constraints (P). To write a constraint for a for a pair of tasks (i, i'), we have

$$X_{ijt} + \sum_{t''=t-\tau_{i'j}-\sigma_{i'ij}+1}^{t-\tau_{i'j}} X_{i'jt''} \leq 1 \quad \forall i, i' \neq i, j, t \tag{6.22}$$

which is (PI), (PII), and (PIII) and will be referred to as constraints (P).

6.1.5. Remarks

The problem in §6.1.1 is a *traditional* scheduling problem which involves only sequencing and timing decisions. In addition to sequencing and timing, the problem in §6.1.2 includes also task-unit assignment decisions, and the problem in §6.1.3 includes also batching decisions. For the problems in §6.1.1 and §6.1.2, each task is to be scheduled once, whereas in the problem of §6.1.3, the number of times that a task is scheduled is determined by the optimization model. Discrete-time MIP models can be applied to all of the aforementioned problem classes, while other methods are limited in the types of problems they can address or the objectives they can handle.

Furthermore, discrete-time models can be easily extended to account for time-varying resource availability and events during the execution of a task (e.g., intermediate material loading), and can be used as a basis for many problems. For example, they can be extended to address problems in production environments where batches can be split apart or mixed (Sundaramoorthy and Maravelias, 2011b), problems with other types of constraints (e.g., limited utilities) (Zyngier and Kelly, 2009; Velez and Maravelias, 2013), as well as problems with various objective functions (Merchan et al., 2016). Thus, our choice to focus on discrete-time MIP formulations means that the proposed changeover constraints will be applicable to a wide range of problems.

6.2. Facet-defining Constraints (SIIT)

Two modifications can be made to constraints (SI). First, many of the inequalities (SI) are redundant, as t' is written for each time point within $\{t - \max\left(\min_{i' \neq i} \tau_{i'j}, \max_{i' \neq i} \sigma_{i'ij}\right), \dots, t - \min_{i' \neq i} \tau_{i'j}\}$. Second, for the remaining necessary constraints, more variables related to i and i' can be added to the left hand side (LHS) so that the constraints are tightened. Based on these two observations, we propose constraints (SIIT), which do not include redundant constraints, while making each constraint as tight as possible. The general form is,

$$\sum_{t'=t-\tau_{ij}+1}^{t} X_{ijt'} + \sum_{i' \neq i} \sum_{t'=t-\omega a_{ijni'}}^{t-\omega b_{ijni'}} X_{i'jt'} \leq 1 \quad \forall i, j, t, n \in \mathbf{N}_{ij} \tag{6.23}$$

where the first summation includes all variables corresponding to processing time points ($t' \in \mathbf{T}_{ijt}^{P}$) for task i, and the second term is a summation over i' and $t' \in \{t - \omega a_{ijni'}, \dots, t - \omega b_{ijni'}\}$

(parameters $\omega a_{ijni'}$ and $\omega b_{ijni'}$ will be defined later). Similar to (SII), we introduce $|\mathbf{N}_{ij}|$ inequalities for every (i,j,t), and each index $n \in \mathbf{N}_{ij}$ is associated with an integer parameter μ_{ijn} (that may be different from the parameter ν_{ijn} used in (SII)). Parameter μ_{ijn} is the largest $\omega b_{ijni'}$ over index i' that appears in the second summation.

Similarly to (SII), μ_{ij1} is set to the smallest processing time of any i',

$$\mu_{ij1} = \min_{i' \neq i} \tau_{i'j} \tag{6.24}$$

The value of $\mu_{ij,n+1}$ is chosen according to (6.25), so that no variables corresponding to changeover time points are excluded in (6.23).

$$\mu_{ij,n+1} = \min\{\mu_{ijn} + \min_{i':\mu_{ijn}<\sigma_{i'ij}} \tau_{i'j}, \max_{i' \neq i} \sigma_{i'ij}\} \tag{6.25}$$

Before specifying $\omega a_{ijni'}$ and $\omega b_{ijni'}$ and thus completing constraints (6.23), we introduce two disjoint task subsets: $\mathbf{IA}_{ijn} = \{i' | i' \neq i, \mu_{ijn} \leq \sigma_{i'ij}\}$ and $\mathbf{IB}_{ijn} = \{i' | i' \neq i, \mu_{ijn} > \sigma_{i'ij}\}$.

Parameters $\omega a_{ijni'}$ and $\omega b_{ijni'}$ depend on the subset task i' belongs to, and are defined as follows,

$$\omega a_{ijni'} = \begin{cases} \tau_{i'j} + \mu_{ijn} - 1, & i' \in \mathbf{IA}_{ijn} \\ \tau_{i'j} + \sigma_{i'ij} - 1, & i' \in \mathbf{IB}_{ijn} \end{cases}$$
$$\omega b_{ijni'} = \begin{cases} \mu_{ijn}, & i' \in \mathbf{IA}_{ijn} \\ \max\{\sigma_{i'ij}, \mu_{ijn} - \min_{i'' \in \mathbf{IA}_{ijn}} \sigma_{i''i'j}, \max_{i'' \in \mathbf{IB}_{ijn} \setminus \{i'\}} (\sigma_{i''ij} - \sigma_{i''i'j})\}, & i' \in \mathbf{IB}_{ijn} \end{cases} \tag{6.26}$$

The definitions in (6.26) are discussed next.

(a) If $i' \in \mathbf{IA}_{ijn}$, there are $\tau_{i'j}$ variables $X_{i'jt'}$ in (6.23), which is the maximum number of variables to be included for a task; including more than $\tau_{i'j}$ i'-indexed variables will make the constraint invalid, because it would cut off the solution in which task i' is carried out back-to-back.

(b) If $i' \in \mathbf{IB}_{ijn}$, the smallest index t' for variables $X_{i'jt'}$ included in the constraint is the earliest changeover time point ($t' = t - \tau_{i'j} - \sigma_{i'ij} + 1$) and the largest t' for included $X_{i'jt'}$ variables must be one period before the smallest of the following three (so that the constraint does not exclude any feasible sequences while remaining as tight as possible):

(b1) t' which would lead to the inclusion of exactly $\tau_{i'j} + 1$ variables for task i' (if $\omega b_{ijni'} = \sigma_{i'ij}$);

(b2) t' that would make (6.23) cut off the solution where $i'' \in \mathbf{IA}_{ijn}$ takes place at $t - \tau_{i''j} - \mu_{ijn} + 1$ and i' takes place $\tau_{i''j} + \sigma_{i''i'j}$ periods later (if $\omega b_{ijni'} = \mu_{ijn} - \sigma_{i''i'j}$);

(b3) t' that would make (6.23) cut off the solution where another $i'' \in \mathbf{IB}_{ijn}$ takes place at $t - \tau_{i''j} - \sigma_{i''ij} + 1$ and i' takes place $\tau_{i''j} + \sigma_{i''i'j}$ periods later (if $\omega b_{ijni'} = \sigma_{i''ij} - \sigma_{i''i'j}$).

In Appendix, we include the proof of correctness of (SIIT), and the algorithm summarizing the procedure for the calculation of $\omega a_{ijni'}$ and $\omega b_{ijni'}$. Table 6.2 gives the data for the example in Figure 6.1, with hollow points illustrating which variables are included in (SIIT). Figure 6.2 presents a simple example illustrating how constraints (SIIT) cut off fractional solutions that are feasible for the LP-relaxation of other formulations.

Table 6.2. Data for the 4-task example.

		$\sigma_{i'i}$ (i' = left, i = top)			
	τ_i	T1	T2	T3	T4
T1	7	0	0	6	5
T2	4	1	0	3	1
T3	8	4	2	0	3
T4	6	2	2	2	0

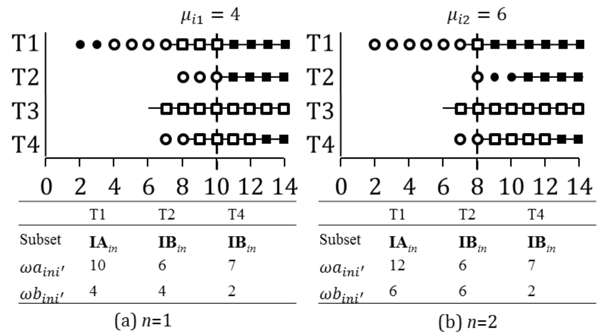


Figure 6.1. Parameters and constraints (SIIT) for i = T3, t = 14, based on the data of Table 6.2.

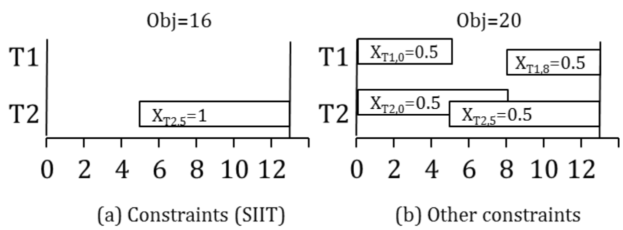


Figure 6.2. Illustration of tightness of constraints (SIIT) for profit maximization. Two tasks, T1 ($\tau_1 = 5$) and T2 ($\tau_2 = 8$), are to be scheduled on a single unit; with changeover times $\sigma_{\text{T1,T2}} = 3$ and $\sigma_{\text{T2,T1}} = 2$; the horizon is 13 hours; and the profit from running T1 and T2 are 4 and 16, respectively. The LP-relaxation with constraints (SIIT) yields the (optimal) integer solution, shown in (a). The solution of the LP-relaxation with all other changeover constraints, shown in (b), is cut off by constraints (21) with $i = \text{T1}, t = 8, n = 1$ ($\omega a_{\text{T1,1,T2}} = 9$, $\omega b_{\text{T1,1,T2}} = 2$).

In Appendix, we include the proof of correctness of (SIIT), and the algorithm summarizing the procedure for the calculation of $\omega a_{ijni'}$ and $\omega b_{ijni'}$. Table 6.2 gives the data for the example in Figure 6.1, with hollow points illustrating which variables are included in (SIIT). Figure 6.2 presents a simple example illustrating how constraints (SIIT) cut off fractional solutions that are feasible for the LP-relaxation of other formulations.

Proposition 6.1 below establishes that (SIIT) are facet-defining for the problem containing constraints (6.7) and (6.23) (proof in Appendix).

Proposition 6.1: Let $H = \{X \in \{0,1\}^{T\cdot|\mathbf{I}|\cdot|\mathbf{J}|}$: *subject to constraints (6.7) (6.23)*$\}$. *Then each inequality in (6.23) is facet-defining for the convex hull of H, conv(H).*

6.3. Relative Tightness of Formulations

We evaluate the relative tightness of the new formulations and the three from the literature. To show that one formulation is at least as tight as another, we prove that any point that is feasible for the LP-relaxation of the tighter formulation is also feasible for the other formulation (see Velez et al., 2017). To show a formulation is tighter, we find a point for a specific instance that is feasible for the less tight formulation, but not for the tighter one. In some cases, neither formulation is tighter, so we find a point that satisfies either one but not the other.

The binary variables included in (K) are for time points after t, while all other changeover constraints include binary variables before t. For consistency, we will use the *backwards* version of (K), henceforth referred to as (KB), where we use M that is the forward analog of constraints (6.13):

$$\sum_{i' \neq i} \sum_{t'=t-\tau_{i'j}-\sigma_{i'ij}+1}^{t-\tau_{i'j}} X_{i'jt'} \leq \left(\sum_{i' \neq i} \sigma_{i'ij}\right) \cdot (1 - X_{ijt}) \quad \forall i, j, t \tag{6.27}$$

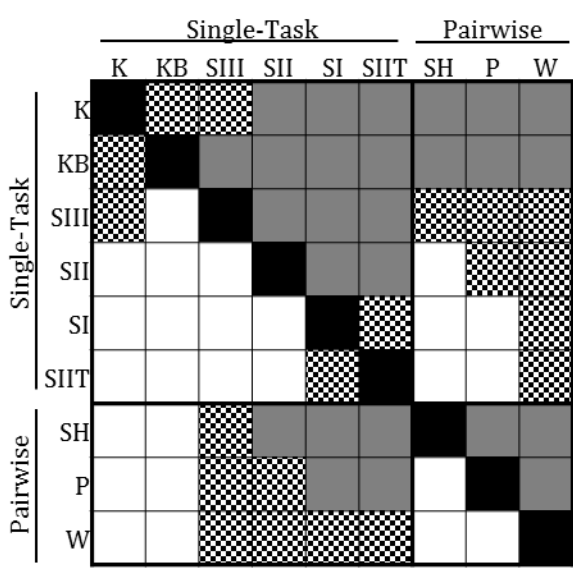


Figure 6.3. Relative tightness of all formulations.

Figure 6.3 summarizes the relative tightness of all formulations. White indicates that the formulation on the left is tighter than the formulation on the top; gray indicates the opposite. A dotted block indicates that neither formulation is tighter. Constraints (W) are the tightest among the pairwise changeover constraints and (SI) and (SIIT) are the tightest among the single-task changeover constraints. In most cases neither the pairwise nor the single-task constraints are tighter. Single-task constraints have the advantage of including binary variables for many different tasks in a single constraint. Pairwise constraints have the advantage that they can include binary variables for more time points for a single task in a single constraint.

6.4. Computational Study

We tested several instances of the three problems we introduced in §6.1: the single unit process, the parallel unit process, and the parallel unit process with unequal capacities. Different objective functions were also studied: tardiness minimization and cost minimization are relatively easier problems, so we considered instances with 5, 10, 15, 20, 25 tasks on 1, 3, 5 units. Makespan minimization is harder, so instances with 7, 8, 9, 10, 11 tasks on 1, 3, 5 units were studied. We used a step length of δ = 1 hour for all instances. Processing times were randomly selected from 3-9 hours (uniform distribution) and rounded up so that they are multiples of the step size, δ, and costs for each task were randomly selected from 1-10 (uniform distribution). For the tardiness minimization instances, due times were randomly selected between zero and the horizon length. For the cost minimization problem with deadlines, deadlines were generated based on the solution of tardiness minimization: if a task in the best solution is finished by the due time, then the due time of this task was used as its deadline; otherwise, we randomly selected the deadline between the finish time of the task and the horizon length. This adjustment was necessary to ensure that the cost minimization instances, with strict deadlines, were feasible.

Changeover times were selected randomly so that they were less than some factor, ε, times the minimum processing time of the two tasks, $\sigma_{i'ij} \in \left[0, \varepsilon \cdot \min\left\{\tau_{ij}, \tau_{i'j}\right\}\right]$. These changeover times may violate the triangle inequality from Assumption 2 (§6.1.4). If $\sigma_{ii''j} > \sigma_{ii'j} + \tau_{i'j} + \sigma_{i'i''j}$, then we chose a new value $\sigma_{ii''j} \in \left[0, \sigma_{ii'j} + \tau_{i'j} + \sigma_{i'i''j}\right]$. We updated the changeover times until all times satisfy the triangular inequality and rounded the changeover times so that they were multiples of the step size, δ. We considered ε = 0.25, 0.5, 0.75 and 1. For a given number of tasks and units, and a given factor, 5 instances were generated. Thus, for makespan and tardiness minimization, we tested 500 instances, and for cost minimization 400 instances.

In this section, we refer to a specific model as $\mathrm{M}_{\mathrm{X}}^{\mathrm{Y}}$, where X denotes the changeover constraints, i.e., X∈{K, SH, W, SI, SII, SIII, SIIT, P}; and Y denotes the objective function, which can be MS (makespan minimization), TRD (tardiness minimization), and CST (cost minimization). We also use MS-VI to denote makespan minimization with valid inequalities (6.28).

$$MS \geq \sum_{i,t} \tau_{ij} X_{ijt} + \sum_{i',t} \left(\min_{i \neq i'} \sigma_{i'ij} \right) X_{i'jt} - \max_{i'} \left(\min_{i \neq i'} \sigma_{i'ij} \right) \quad \forall j \tag{6.28}$$

For example, $\mathrm{M}_{\mathrm{SI}}^{\mathrm{MS}}$ is the model that includes constraints (SI) for makespan minimization. We use M_{X} to denote all models with constraints X, regardless of the objective function; and M^{Y} to denote all models with objective function Y, regardless of changeover constraints.

All the instances were solved using CPLEX 12.6.3 via GAMS 24.7.1 on a cluster with 21 Intel Xeon (E5520) processors at 2.27 GHz and 16 GB of RAM running on CentOS Linux 7, with a 1800-second resource limit. To better assess the effectiveness of the formulations we turned off the aggregator, presolver and presolver for initial relaxation by setting CPLEX options aggind, preind, and relaxpreind to zero. This set of CPLEX options is referred to as C1. Using CPLEX default settings, referred to as C2, requires less time, but leads to similar conclusions, which are briefly discussed in §6.4.

Finally, for constraints (K), we used a tight big-M, calculated using expression (6.29), though the performance is similar to the performance using M defined in expression (6.14).

$$M_{ij} = \sum_{i' \neq i} \left\lceil \frac{\sigma_{ii'j}}{\tau_{i'j}} \right\rceil \tag{6.29}$$

6.4.1. Single Unit

The time horizon, η, is selected to ensure that there is enough time for all tasks to be completed. The calculation is based on the number of tasks and the longest processing time.

$$\eta = (\max_i \tau_i) \cdot (2|\mathbf{I}| - 1) \tag{6.30}$$

The second term in the product is the number of tasks plus changeovers that take place. We multiplied this term by the maximum processing time.

The performance charts in Figure 6.4 show the fraction of instances that are solved by each model within the given amount of time (normalized to the fastest model for each instance). Figure 6.4(a) and 6.4(c) show the results for makespan minimization and tardiness minimization, respectively, using different changeover constraints (SI), (SII), (SIII), (P), (SIIT), (K), (SH), (W). We observe that although $\mathrm{M_W}$ and $\mathrm{M_{SIIT}}$ are the tightest models, they are also the slowest. Also, $\mathrm{M_{SIII}}$ is one of the fastest models although it is among the weakest. Models $\mathrm{M_K}$, $\mathrm{M_{SI}}$, $\mathrm{M_{SII}}$, $\mathrm{M_{SIII}}$, whose changeover constraints are all written in terms of a single task, are much faster than $\mathrm{M_P}$, $\mathrm{M_{SH}}$, $\mathrm{M_W}$, whose changeover constraints are written in terms of pairs of tasks. Figure 6.4(b) shows the results for makespan minimization with the four fastest changeover constraints, (SIII), (SII), (SI) and (K), and with valid inequalities (6.28) included. Adding (6.28) does not lead to computational enhancements for single-unit problems. For makespan minimization, $\mathrm{M_{SII}^{MS}}$ and $\mathrm{M_{SIII}^{MS}}$ are the fastest models; while for tardiness minimization, $\mathrm{M_K^{TRD}}$ is the fastest.

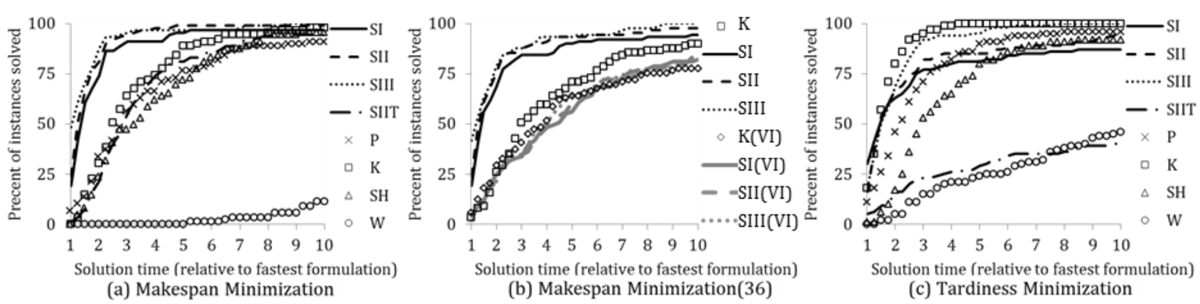


Figure 6.4. Performance charts of different models for single unit instances using C1 set of options.Models with the same objective but different changeover constraints are compared in each sub-plot: (a) makespan minimization; (b) makespan minimization with valid inequalities (6.28); and (c) tardiness minimization. The changeover constraints are shown in the legends.

All models have the same number of variables. Figure 6.5, in logarithmic scale, summarizes the data for the number of constraints, nonzeros, and nodes in the branch-and-bound tree. The models with pairwise changeover constraints, M_P, M_{SH}, M_W, have about an order of magnitude more constraints than the models with single-task constraints, potentially explaining why they are slower. The tightest models, M_W and M_{SIIT}, have the most non-zeros, the number of which is more than 5 times greater than that of the fastest model M_{SIII}. The changeover constraints account for more than 99% of the total constraints for models with pairwise changeover constraints, and for 91% - 95% of the constraints for the models with single-task constraints. With CPLEX setting C1, we find that M_{SI} leads to the smallest branch-and-bound tree, while M_K being the least tight model has the largest tree, with an order of magnitude more nodes searched compared to M_{SI}. If we also turn off the cut generation process in CPLEX, then the number of nodes increases for all models; more importantly, we find that the number of nodes for other models is more than 5 times larger than that for M_{SIIT}, which is expected from the theoretical tightness.

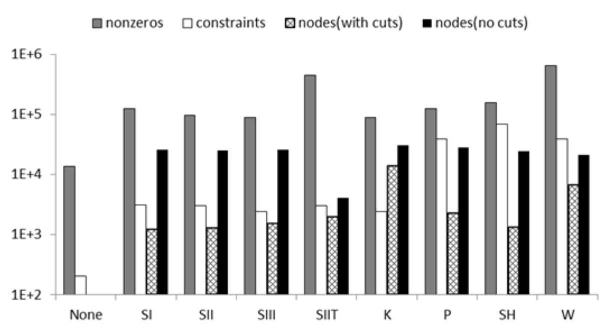


Figure 6.5. Average over all single unit instances of the number of constraints, nonzeros, and branch-and-bound nodes. *None* refers to the model without changeovers.

Table 6.3 presents the average improvement of the integrality gap with respect to the gap of M_K, which has the largest gap (the integrality gaps of any two models can be compared using the relative gap with respect to M_K). Based on these results, models M_{SIIT} and M_W have the greatest integrality gap improvement, followed by M_{SI}, M_{SII}, M_P, M_{SIII}, and then by M_{SH}, an observation that is in agreement with the theoretical study in the previous section. Note that the gap for tardiness minimization varies across changeover constraints much more than that for makespan minimization.

Table 6.3. Single-unit problem: average integrality gap improvement with respect to the gap of $\mathrm{M_K}$.

Problem	K (ref)	SI	SII	SIII	SIIT	P	SH	W
Makespan	0.94	0%	0%	0%	0.04%	0%	0%	0%
Tardiness	0.28	12.10%	12.0.%	10.42%	36.14%	11.24%	9.90%	33.76%

6.4.2. Parallel Units

The time horizon was selected based on the number of tasks, the number of units and the longest processing time as follows,

$$\eta = (\max_{i,j} \tau_{ij}) \cdot \left\lfloor \frac{2|\mathbf{I}| - 1}{|\mathbf{J}|} \right\rfloor \tag{6.31}$$

The term in the round-down operator is the maximum number of tasks plus changeovers that we would expect to take place in a single unit; note that we only need the horizon to be long enough to start the last task, so we rounded down. We multiplied this term by the maximum processing time of any task.

Figure 6.6 shows performance charts for makespan minimization with and without valid inequalities (6.28), for tardiness minimization, and for cost minimization. As in the single-unit problems (§6.4.1), we see that the models with single-task changeover constraints are faster. $\mathrm{M_{SII}^{MS}}$ is the fastest among $\mathrm{M^{MS}}$. We tested the models with the four fastest changeover constraints together with valid inequalities (6.28), and see improvements as expected. With valid inequalities (6.28), $\mathrm{M_{SII}^{MS-VI}}$, $\mathrm{M_{SIII}^{MS-VI}}$, $\mathrm{M_{K}^{MS-VI}}$ are the fastest among $\mathrm{M^{MS-VI}}$, and they are faster than $\mathrm{M^{MS}}$. For tardiness minimization, $\mathrm{M_{K}^{TRD}}$ is the fastest; while for cost minimization, $\mathrm{M_{SII}^{CST}}$ and $\mathrm{M_{SIII}^{CST}}$ are the fastest. Table 6.4 shows the average integrality gap improvement for different models with respect to $\mathrm{M_K}$. Similar to the results in §6.1.4, $\mathrm{M_{SIIT}}$, has the greatest integrality gap reduction.

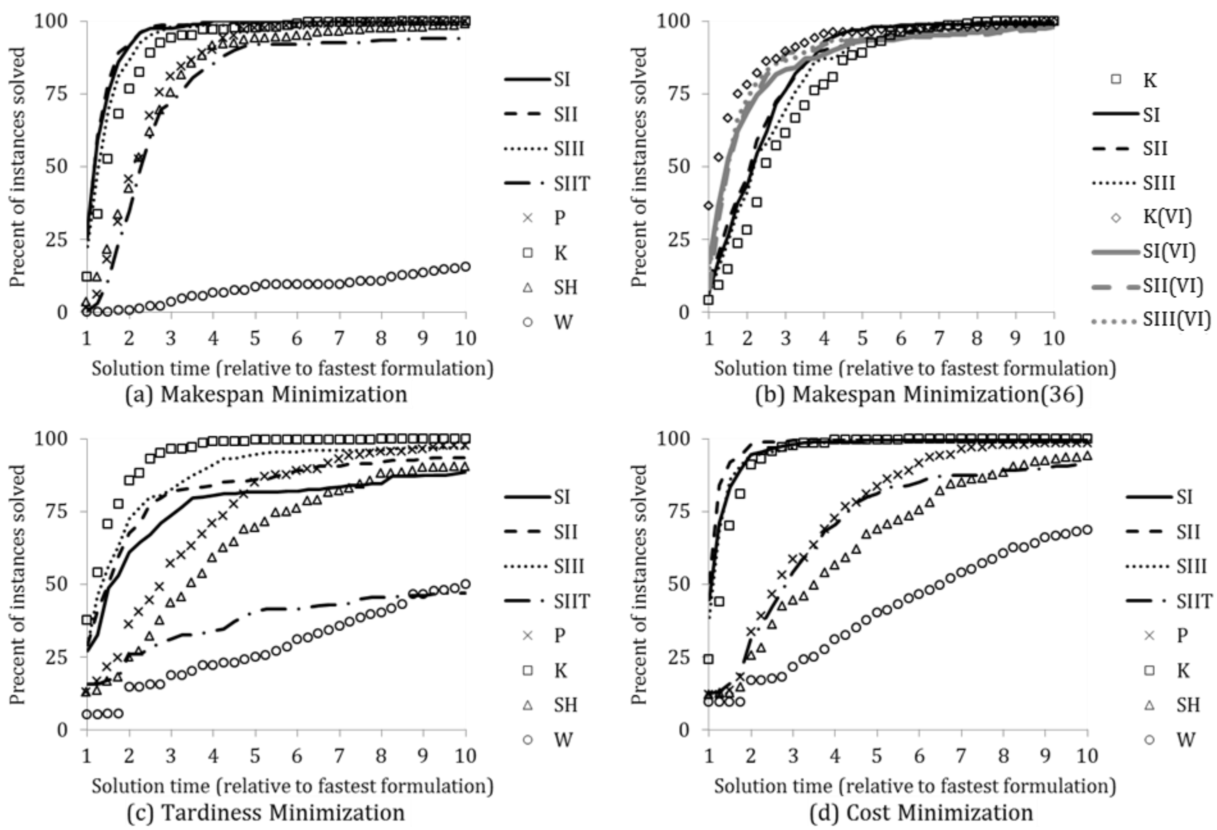


Figure 6.6. Performance charts of different models for instances with parallel units using C1 set of options.

Table 6.4. Parallel units problem: average integrality gap improvement with respect to the gap of M_K.

Problem	K(ref)	SI	SII	SIII	SIIT	P	SH	W
Makespan	0.91	0%	0%	0%	0.06%	0%	0%	0.03%
Tardiness	0.10	12.03%	12.01%	11.23%	24.77%	11.53%	10.32%	19.89%
Cost	0.04	29.78%	29.78%	29.78%	36.72%	29.51%	29.13%	36.42%

6.4.3. Parallel Units with Unequal Capacities

Unit capacities were randomly selected with a uniform distribution from 10-30 kg, and customer demands were randomly selected from 20-40 kg. The time horizon for the makespan minimization instances was chosen as

$$\eta = \left\{\max_{i,j}\left[\tau_{ij}\left\lceil \xi_i/\beta_j \right\rceil\right]\right\} \cdot \left\lfloor \frac{2|\mathbf{I}|-1}{|\mathbf{J}|} \right\rfloor \qquad (6.32)$$

The first term is the maximum total time any task needs to run for to meet demand, which we found by multiplying the processing time by the number of runs required in each unit and taking the maximum over all units. The second term is the number of tasks plus changeovers expected on a single unit.

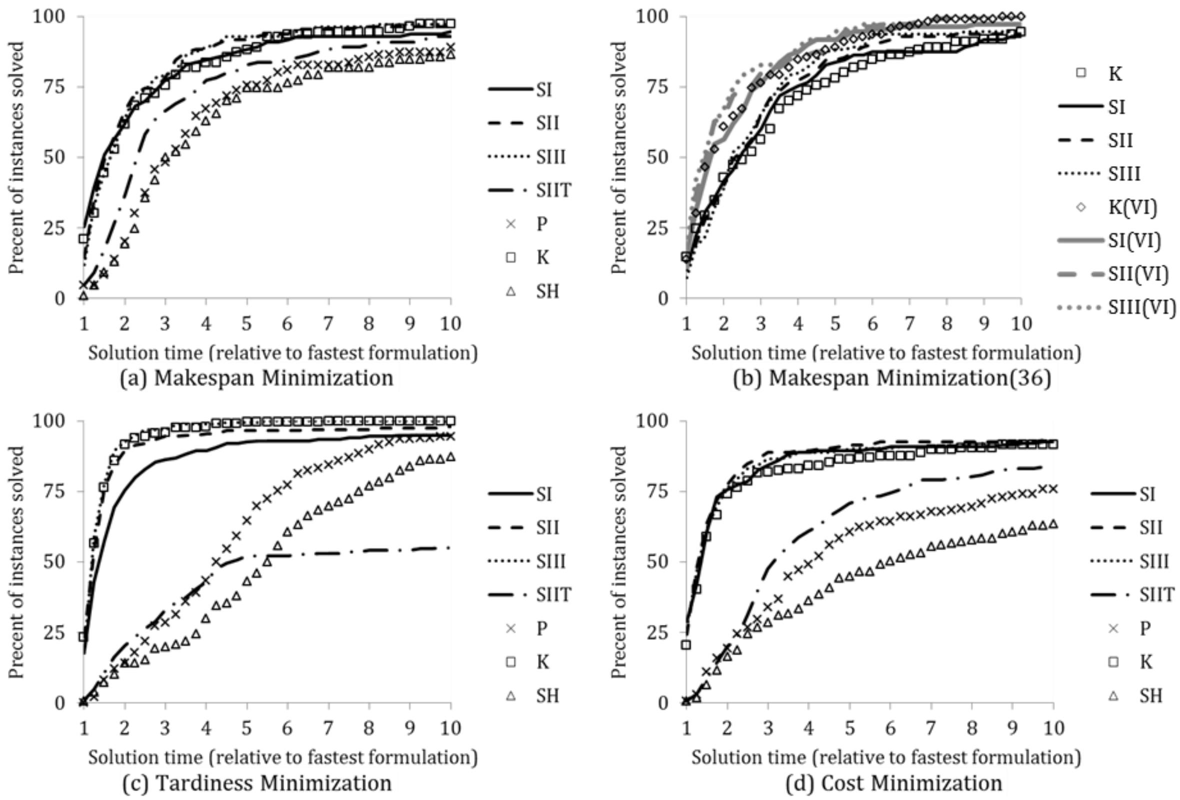


Figure 6.7. Performance charts of different models for unequal capacity parallel units instances using C1.

Figure 6.7 shows the performance charts. Constraints (W) are not tested because they are not applicable when a task can be executed multiple times. For makespan minimization, M_{SII}^{MS}, M_{SIII}^{MS}, M_{K}^{MS} are the fastest, while M_{SIIT}^{MS} and the two models with pairwise changeover constraints M_{P}^{MS} and M_{SH}^{MS} are the slowest. Same as in the parallel units problems (§6.4.2), adding valid inequalites (6.28) significantly improves the solution times for makespan minimization. For tardiness minimization, M_{SIII}^{TRD} and M_{K}^{TRD} are the fastest, while for cost minimization, M_{SII}^{CST} and M_{SIII}^{CST} are the most fastest. Table 6.5 compares the improvement of integrality gaps of the models.

Table 6.5. Parallel units with unequal capacities: average integrality gap improvement with respect to M_K.

Problem	K(ref)	SI	SII	SIII	SIIT	P	SH
Makespan	0.94	0%	0%	0%	0.04%	0%	0%
Tardiness	0.35	0.94%	0.94%	0.87%	2.69%	0.82%	0.80%
Cost	0.26	3.43%	3.43%	3.21%	5.18%	3.19%	3.10%

6.4.4. *Additional Testing*

We also studied the performance of all models using CPLEX default settings (C2). The relative performance of the different changeover constraints remains the same, though solution times are shorter. Table 6.6 summarizes how valid inequalities (6.28) affect the solution time for makespan minimization problems, using C1 or C2. For the single unit problem, the average solution time using valid inequalities (6.28) and C1 is 1%-106% higher, though it decreases by 36%-48% when C2 settings are used. For the parallel units problem, the average solution time using valid inequalities (6.28) decreases by 1%-29% when C1 is used and by 39%-48% when C2 is used. For the instances with unequal capacity units, solution times using valid inequalities (6.28) decrease by 18%-33% (10%-13%) when using C1 (C2). We also tested all instances with both preprocessing and cut generation turned off. The computational times in this case increase significantly: only 63% of the instances were solved to optimality within 3600 seconds.

Table 6.6. Solution times for makespan minimization problems (different formulations and valid inequalities).

Problem	CPLEX	K	SI	SII	SIII	K(36)	SI(36)	SII(36)	SIII(36)
Single unit	C1	1.00	0.49	0.48	0.57	1.01	1.01	0.96	0.93
	C2	0.31	0.33	0.32	0.34	0.20	0.17	0.16	0.19
Parallel units	C1	1.00	0.94	0.94	0.99	0.71	0.93	0.93	0.96
	C2	0.55	0.79	0.77	0.73	0.34	0.41	0.41	0.40
Parallel units with unequal capacities	C1	1.00	1.12	1.00	0.99	0.82	0.75	0.73	0.71
	C2	0.31	0.34	0.34	0.35	0.27	0.31	0.31	0.30

Note. The solution time of each instance is normalized with respect to the solution time for M_K^{MS} using CPLEX setting C1. We use the average normalized computational time over all instances for the same type of problem.

6.5. Conclusions

In this chapter, we proposed one new formulation for sequence-dependent changeover times in discrete-time MIP scheduling models, and compared them with previous formulations both theoretically and computationally. In terms of tightness, (SI), (SII), (SIIT) and (P) are tighter than the two literature constraints, (SH) and (K). Among the constraints written for a single task, constraints (K) are the weakest, followed by (SIII) and (SII); and (SI) and (SIIT) are the tightest. Among the constraints written for pairs of tasks, constraints (W) are the tightest, and (SH) are the weakest. Constraints (W), however, cannot be applied to problems in which a task can be executed more than once. In terms of computational effectiveness using CPLEX, although models with constraints (K), (SIII), (SII), and (SI) have similar solution times, models with constraints (SII) and (SIII) are typically the fastest. We observed that tighter formulations did not necessarily lead to faster computational times, as tighter formulations had typically more constraints and/or non-zeros, and took longer time to solve the LP relaxations. Also, models with changeover constraints written for a single task were faster than those written for pairs of tasks. We observe similar computational results using Gurobi.

The constraints presented in this chapter can be added to any discrete-time MIP scheduling formulation, including models developed to address problems in complex production environments (e.g., environments with batch mixing, splitting and recycling) as well as problems with a range of processing characteristics (e.g., general resource constraints, time-varying resource availability and cost). Thus, the constraints presented herein are relevant for many real-world problems.

6.6. Notation

Indices/Sets

$i \in \mathbf{I}$	Tasks (jobs)
$j \in \mathbf{J}$	Units (machines)
$t \in \mathbf{T}$	Time points/periods
$n \in \mathbf{N}$	Index used to select times points when a constraint is written

Subsets

$\mathbf{I}_j$	Tasks that can be processed in unit j
$\mathbf{J}_i$	Units that can process task i
$\mathbf{N}_{ij}$	Indices n for which ν_{ijn} and μ_{ijn} are defined
$\mathbf{T}_{ijt}^{P}$	$= \{t' \mid t - \tau_{ij} + 1 \leq t' \leq t\}$, for given (i, j, t)
$\mathbf{T}_{i'ijt}^{C}$	$= \{t' \mid t - \tau_{i'j} - \sigma_{i'ij} + 1 \leq t' \leq t - \tau_{i'j}\}$, for given (i, i', j, t)

Parameters

α_{ij} Cost to run task i in unit j

β_j Capacity of unit j

δ Time step

ε Factor used to determine the maximum changeover length

η Time horizon

ν_{ijn}/μ_{ijn} Parameter to select the time points for which (SII)/(SIIT) is written for task i and unit j

ξ_j Demand for the output of task i

$\sigma_{ii'j}$ Changeover time between task i and task i' on unit j

τ_{ij} Fixed processing time for task i in unit j

$\phi_i/\bar{\phi}_i$ Due time/dealine of task i

$\omega a_{ijni'}/\omega b_{ijni'}$ Parameters to define the summation over i' for given (i, j, n) in (SIIT)

Binary Variables

X_{ijt} $= 1$ if and only if task i starts on unit j at time point t

Continuous Nonnegative Variables

CST Total cost

MS Makespan

TRD_i Tardiness for task i

TRD Total tardiness

Chapter 7

Conclusions and Recommendations

7.1. Concluding Remarks

In this thesis, we studied inventory routing problem from different aspects. MIP models considering different constraints were proposed, and solution methods were developed in order to solve realistic instances in a timely fashion. We evaluated different policies for reoptimizing a maritime IRP under uncertainty. Furthermore, we explored two research topics related to IRP.

First, we proposed MIP models that can account for a wide range of constraints, which are necessary for obtaining an implementable solution. The models are based on a discrete-time approach and time-expanded network representation. The complex constraints include maximum daily working and driving time, driver resting and checking-in/out, time varying consumption rate, and multiple access windows. However, the proposed model leads to prohibitively long solution time for larger instances. A network with more than eight customers cannot be solved in a relistic time frame (less than an hour).

Second, we developed solution methods, which lead to solutions of high quality obtained in a reasonable time. A preprocessing algorithm reduces the nodes and the arcs in the distribution network, based on the current inventory and forecast consumption profile. After preprocessing, only the customers with demands that should be fulfilled in the planning horizon and their nearby customers are included. A decomposition algorithm iteratively solves an upper level vehicle routing subproblem and a lower level detailed scheduling subproblem. In the upper level, a MIP model is solved to select the optimal routes and the corresponding trucks to carry out the routes, and the distribution cost is minimized. In the lower level, drivers are explicitly modeled, and a continuous-time approach is used to obtain a detailed schedule, based on the upper level decisions. Different options for running the iterative algorithm were presented. Using the proposed algorithm, instances with distribution networks including up to 155 customers (34 customers after preprocessing) were solved within half an hour.

Third, we developed a framework for the reoptimization of maritime IRP under uncertainty, based on MIP models and stochastic simulations. The MIP model is formulated on a discrete-time approach, and considers vessels in long- and short-term renting modes, and owned and third-party production nodes. The stochastic simulations consider uncertainty in vessel availability, trip delays, production/consumption variations, and pick-up windows. We showed that even when no uncertainty is incorporated, the closed-loop solution (i.e., the implemented solution in a rolling horizon manner) is very different from the open-loop solution (i.e., the initial solution obtained from the optimization model). When uncertainty is incorporated, the closed-loop cost increases by 30%. We also identified policies which lead to high quality closed-loop solutions.

Fourth, we developed novel terminal constraints for online scheduling. Different network structures were considered, including multi-stage single-product, single-stage multi-product, and multi-stage multi-product. The proposed terminal constraints can prevent stock out, as well as save inventory holding cost, compared to (1) the model without any terminal constraints and (2) the model using a traditional threshold approach. Furthermore, for two types of networks, we proved that the terminal constraints can lead to recursive feasibility.

Finally, we proposed one new formulation for modeling sequence dependent changeover in scheduling problems, and we proved that the proposed formulation is facet-defining for a certain problem. Moreover, we compared the proposed formulation with seven formulations that were previously developed, both theoretically and computationally. Interestingly, the tighter formulations can lead to longer solution times, because the solution time of their LP relaxations is longer.

7.2. Future Research Directions

First, we have made certain assumptions when developing the IRP models and solution methods. It will be interesting to see how one can modify the models and solution methods if the assumptions are not satisfied.

(a) We assumed that drivers are dedicated to products, so that we can solve the distribution problems of each product independently. However, drivers can be shared among products in reality. To consider this, the distribution of different products should be determined simultaneously, which will lead to very large models. Therefore, one more layer might be needed in the solution method, which decides the driver assignment to different products.

(b) We assumed that there is no limit on the amount of products at the plant. However, this assumption might not be satisfied, because the production scheduling and IRP are not entirely decoupled problems. To consider the production schedules, we may need to develop and solve a MIP model in the dynamic network reduction phase (for a longer horizon) to decide the customers to visit (in a relatively short horizon).

(c) We assumed that there is only one plant in the supply chain. For the supply chains including multiple plants, the MIP model should be modified, to consider that a route can start from one plant but end at another.

Second, we can further study how reoptimization should be conducted for the vehicle-based IRP.

(a) In the dynamic network reduction algorithm, we used the safety stock level as a terminal constraint to decide the minimum demand in the planning horizon. However, whether the safety stock would be the "optimal" terminal constraint for reoptimization is unknown.

(b) As shown in Chapter 4 for MIRP, the closed-loop solution might be very different from the open-loop solution, and one can expect that this is also true for vehicle-based IRP. Therefore, how to obtain high-quality closed-loop solution is an open question. Possible directions include the use of certain policies and the modification of objective function to reflect the long-term effect.

Finally, there are a few interesting research extensions related to the terminal constraints and changeovers.

(a) For the terminal constraints we proposed in Chapter 5, three types of network structures were studied. It would be interesting to find out how to write terminal constraints for more complex networks, which include mixing or recycling operations.

(b) In Chapter 6, we assumed that there is no cost associated with changeovers. Therefore, we can further study what would be a good formulation when changeover cost should be considered.

Appendices

A. Proof of Proposition 5.1

Proposition 5.1*:* Let $L_{mt}(S1)$ and $W_{ijt}(S1)$ be the values from a feasible solution, $S1$, of model MF. If model MF has a feasible solution, $S2$, when using $S_m = L_{m,T+1}(S1) + \delta_m + \sum_{i\in\mathbf{I}_m^-, j\in\mathbf{J}_i} \beta_{ij} W_{ij,T}(S1)$, then for any $1 < \sigma \leq T+1$, using $S_m = L_{m,\sigma}(S1) + \delta_m + \sum_{i\in\mathbf{I}_m^-, j\in\mathbf{J}_i} \beta_{ij} W_{ij,\sigma-1}(S1)$, model MF also has a feasible solution, $S3$.

Proof: Given solution $S1$ and $S2$, we can construct a schedule of time $\{0, \ldots, 2T\}$, denoted by inventory level variable $L_{mt}(2T)$ and task start variable $W_{ijt}(2T)$. We show that based on this schedule, we can find a feasible solution $S3$.

We use $L_{mt}(S1)/L_{mt}(S2)$ and $W_{ijt}(S1)/W_{ijt}(S2)$ to denote the variable values of solution $S1/S2$. Since $S_m = L_{m,T+1}(S1) + \delta_m + \sum_{i\in\mathbf{I}_m^-, j\in\mathbf{J}_i} \beta_{ij} W_{ij,T}(S1)$ for solution $S2$, we can construct the schedule of $\{0, \ldots, 2T\}$ as follows:

$$L_{mt}(2T) = L_{mt}(S1) \quad \forall m, t \leq T$$

$$L_{mt}(2T) = L_{m,t-T}(S2) \quad \forall m, T < t \leq 2T$$

$$W_{ijt}(2T) = W_{ijt}(S1) \quad \forall i, j, t \leq T - \tau_{ij}$$

$$W_{ijt}(2T) = 0 \quad \forall i, j, T - \tau_{ij} + 1 \leq t \leq T - 1$$

$$W_{ijt}(2T) = W_{ij,t-T}(S2) \quad \forall i, j, T \leq t \leq 2T$$

These variables $L_{mt}(2T)$ and $W_{ijt}(2T)$ satisfy all constraints (5.1b)-(5.1d) after modifying the time domain of the constraints to $0 < t \leq 2T$.

Thus, for any $1 < \sigma \leq T+1$, when $S_m = L_{m,\sigma}(S1) + \delta_m + \sum_{i\in\mathbf{I}_m^-, j\in\mathbf{J}_i} \beta_{ij} W_{ij,\sigma-1}(S1)$, we can construct a feasible solution, $S3$, for model MF, whose variables are given as follows,

$$L_{mt}(S3) = L_{m,t+\sigma-1}(2T) \quad \forall m, 0 < t \leq T$$

$$W_{ijt}(S3) = W_{ij,t+\sigma-1}(2T) \quad \forall i, j, 0 \leq t \leq T$$

Since constraints (5.1b)-(5.1d) are satisfied for $0 < t \leq 2T$, $S3$ is a feasible solution for MF. ■

B. Proof of Proposition 5.2

Proposition 5.2: For multi-stage single-product problems, model MF is always feasible regardless of the horizon length, if initial inventory levels S_k satisfy constraints (5.3).

Proof: Based on the assumption that model MC is feasible, we know that $\beta_k \geq \tau_k \delta$, for every stage k. We use $S_k(t) \in \mathbb{R}^+$ to denote the inventory level of the material that has been produced in stage k at time t, without considering the consumption of the final product. We use $S_k^+(t)$ to denote the material that is being produced in stage k at time t; it can only be 0 or β_k. Because there is no consumption of the product considered in $S_k(t)$, the following inequalities (B1) are obvious,

$$\sum_{k'=k}^{|\mathbf{K}|} S_{k'}(t') + \sum_{k'=k+1}^{|\mathbf{K}|} S_{k'}^+(t') \geq \sum_{k'=k}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=k+1}^{|\mathbf{K}|} S_{k'}^+(t) \quad \forall k, t, t' > t \tag{B1}$$

We show that with constraints (5.3), there exists a schedule satisfying $S_{|\mathbf{K}|}(t) \geq t\delta$ for any $t \in \mathbb{N}$; based on this schedule, we can construct a feasible solution for model MF, and therefore prove Proposition 5.2.

First, we prove the following lemma by mathematical induction.

Lemma B1: If constraints (5.3) are satisfied, then there exists a schedule satisfying inequalities (B2) below.

$$\sum_{k'=k}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=k+1}^{|\mathbf{K}|} S_{k'}^+(t) \geq t\delta + \left(\sum_{k'=k+1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k+1}^{|\mathbf{K}|} \beta_{k'} \quad \forall k, t \tag{B2}$$

When $k = 1$, we know that if the task in stage 1 keeps running, then the total inventory levels of all stages (without consuming the product) is greater than the initial value from the inequality of k=1 of (5.3), plus the production in stage 1, as follows,

$$\sum_{k'=1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=2}^{|\mathbf{K}|} S_{k'}^+(t) \geq \left(\sum_{k'=1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=2}^{|\mathbf{K}|} \beta_{k'} + \left\lfloor \frac{t}{\tau_1} \right\rfloor \beta_1 \quad \forall t$$

$$\Rightarrow \sum_{k'=1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=2}^{|\mathbf{K}|} S_{k'}^{+}(t) \geq \left(\sum_{k'=1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=2}^{|\mathbf{K}|} \beta_{k'} + \left(\frac{t}{\tau_1} - 1\right) \beta_1 \quad \forall t \tag{B3}$$

Since $\beta_1 \geq \tau_1 \delta$, we have

$$\sum_{k'=1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=2}^{|\mathbf{K}|} S_{k'}^{+}(t) \geq \left(\sum_{k'=1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=2}^{|\mathbf{K}|} \beta_{k'} + \left(\frac{t}{\tau_1} - 1\right) \tau_1 \delta \quad \forall t$$

$$\Rightarrow \sum_{k'=1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=2}^{|\mathbf{K}|} S_{k'}^{+}(t) \geq \left(\sum_{k'=1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=2}^{|\mathbf{K}|} \beta_{k'} + (t - \tau_1)\delta \quad \forall t$$

$$\Rightarrow \sum_{k'=1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=2}^{|\mathbf{K}|} S_{k'}^{+}(t) \geq t\delta + \left(\sum_{k'=2}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=2}^{|\mathbf{K}|} \beta_{k'} \tag{B4}$$

Therefore, (B2) of $k = 1$ is satisfied.

Now, assuming (B2) of $k = k1$ is satisfied, we show that (B2) of $k = k1 + 1$ is also satisfied; i.e., we prove that if (B5) below is satisfied, (B6) is also satisfied.

$$\sum_{k'=k1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}^{+}(t) \geq t\delta + \left(\sum_{k'=k1+1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k1+1}^{|\mathbf{K}|} \beta_{k'} \tag{B5}$$

$$\sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t) \geq t\delta + \left(\sum_{k'=k1+2}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k1+2}^{|\mathbf{K}|} \beta_{k'} \tag{B6}$$

Apparently, if at t inventory levels satisfy (B7) below, (B6) is also satisfied, since each batch takes at least one period. We can see from (5.3) that (B7) is satisfied for $t = 0$.

$$\sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t) \geq (t-1)\delta + \left(\sum_{k'=k1+1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k1+2}^{|\mathbf{K}|} \beta_{k'} \tag{B7}$$

Now, we show that in the case that (B7) is violated, (B6) is also satisfied. Let $t1$ denote the first time point (in its neighborhood) that (B7) is violated, i.e., (B7) is satisfied for $t = t1 - 1$, but violated for $t = t1$. Mathematically, they are

$$\sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t1-1) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t1-1) \geq (t1-2)\delta + \left(\sum_{k'=k1+1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k1+2}^{|\mathbf{K}|} \beta_{k'} \qquad \text{(B8)}$$

$$\sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t1) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t1) < (t1-1) \cdot \delta + \left(\sum_{k'=k1+1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k1+2}^{|\mathbf{K}|} \beta_{k'} \qquad \text{(B9)}$$

From (B1) and $\beta_{k1+1} \geq \tau_{k+1}\delta$, we know that the LHS of (B8) and (B9) are equal, i.e.,

$$\sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t1-1) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t1-1) = \sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t1) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t1) \qquad \text{(B10)}$$

From $t = t1 - 1$ of (B5) and (B9), (B10), we know that,

$$S_k(t1-1) + S_{k+1}^{+}(t1-1) > \beta_{k1+1} \qquad \text{(B11)}$$

Thus, at time interval $[t1, t1 + \tau_{k1+1} - 1]$, one more batch can be finished in stage $k1$+1. (B11) together with (B8) leads to

$$\sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t1+\tau_{k1+1}-1) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t1+\tau_{k1+1}-1)$$
$$\geq (t1-2)\delta + \left(\sum_{k'=k1+1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k1+2}^{|\mathbf{K}|} \beta_{k'} + \beta_{k1+1}$$

Since $\beta_{k1+1} \geq \tau_{k1+1}\delta$, we have

$$\sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t1+\tau_{k1+1}-1) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t1+\tau_{k1+1}-1)$$
$$\geq (t1+\tau_{k1+1}-2)\delta + \left(\sum_{k'=k1+1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k1+2}^{|\mathbf{K}|} \beta_{k'} \qquad \text{(B12)}$$

which means that (B7) of $t = t1 + \tau_{k1+1} - 1$ is satisfied again. Now, we only need to show for any time $t \in [t1, t1 + \tau_{k1+1} - 2]$, (B6) is satisfied. For any $t \in [t1, t1 + \tau_{k1+1} - 2]$, from (B1) and (B8), we know

$$\sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t) \geq \sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t1-1) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t1-1)$$

$$\geq (t1-2)\delta + \left(\sum_{k'=k1+1}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k1+2}^{|\mathbf{K}|} \beta_{k'}$$

$$\Rightarrow \sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t) \geq (t1+\tau_{k1+1}-2)\delta + \left(\sum_{k'=k1+2}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k1+2}^{|\mathbf{K}|} \beta_{k'}$$

$$\Rightarrow \sum_{k'=k1+1}^{|\mathbf{K}|} S_{k'}(t) + \sum_{k'=k1+2}^{|\mathbf{K}|} S_{k'}^{+}(t) \geq t\delta + \left(\sum_{k'=k1+2}^{|\mathbf{K}|} \tau_{k'}\right) \cdot \delta + \sum_{k'=k1+2}^{|\mathbf{K}|} \beta_{k'}$$

which is (B6).

Therefore, we have shown that if (B2) of $k = k1$ is satisfied, (B2) of $k = k1 + 1$ is also satisfied, which finishes the proof of Lemma B1 by mathematical reduction.

Making the stage $k = |\mathbf{K}|$ in Lemma B1, we have

$$S_{|\mathbf{K}|}(t) \geq t\delta \quad \forall t$$

Accordingly, based on $S_k(t)$, we can construct a feasible solution of MF, as follows,

$$W_{kt} = 1 \text{ if and only if } \sum_{k'=k}^{|\mathbf{K}|} S_{k'}(t+\tau_{k'}) + \sum_{k'=k+1}^{|\mathbf{K}|} S_{k'}^{+}(t+\tau_{k'})$$

$$> \sum_{k'=k}^{|\mathbf{K}|} S_{k'}(t+\tau_{k'}-1) + \sum_{k'=k+1}^{|\mathbf{K}|} S_{k'}^{+}(t+\tau_{k'}-1) \quad \forall k, t$$

$$L_{kt} = S_k(0) + \beta_k \sum_{t'<t-\tau_k} W_{kt'} - \beta_{k+1} \sum_{t'<t} W_{k+1,t'} \quad \forall k < |\mathbf{K}|, t$$

$$L_{|\mathbf{K}|,t} = S_{|\mathbf{K}|}(0) + \beta_{|\mathbf{K}|} \sum_{t'<t-\tau_{|\mathbf{K}|}} W_{|\mathbf{K}|,t'} - t\delta \quad \forall t$$

Therefore, when constraints (5.3) are satisfied, model MF is always feasible. ■

C. Proof of Proposition 5.4

Proposition 5.4: For single-stage multi-product problems, model MF is always feasible regardless of the horizon length, if initial inventory levels S_i satisfy constraints (5.6).

Proof: We show that if initial inventory levels satisfy constraints (5.6), there exists a schedule that can satisfy demand for any $t \in \mathbb{N}$; based on this schedule, we can construct a feasible solution for model MF, and therefore prove proposition 5.4.

We use $S_i(t) \in \mathbb{R}^+$ to denote the inventory level of product i at time t. To show there exists a schedule that can satisfy demand for any $t \in \mathbb{N}$, we show that if the inventory levels $S_i(t)$ satisfy constraints (5.6) and the unit is idle at t, one can always start a certain task $i1$ to process. During the process, inventory levels are non-negative, and after the process, inventory levels $S_i(t + \tau_{i1})$ satisfy constraints (5.6) again.

Since the initial inventory levels S_i satisfy constraints (5.6), we know that at time $t = 0$, (5.6) is satisfied and the unit is idle.

Consider a time t at which the unit is idle. Without loss of generality, we assume that $S_{i1}(t)$ satisfy

$$\frac{S_{i1}(t)}{\delta_{i1}} = \min_i \frac{S_i(t)}{\delta_i} \tag{C1}$$

If the following constraints are satisfied,

$$\sum_{i \in \mathbf{I}_p} \frac{c_i \tau_i S_i(t)}{\delta_i} \geq \left(\sum_{i \in \mathbf{I}_p} c_i \tau_i \right) \cdot \left(\sum_{i \in \mathbf{I}_p} \tau_i \right), \forall p \in \mathbf{P}(\mathbf{I}) \tag{C2}$$

Then, we can process task $i1$.

It is easy to show that inventory levels are non-negative during the process of task $i1$, because for all i, $S_i(t)/\delta_i \geq S_{i1}(t)/\delta_{i1}$, and $S_{i1}(t)/\delta_{i1} \geq \tau_1$ from the inequality (C2) with $p = \{i1\}$.

To show inventory levels $S_i(t + \tau_{i1})$ satisfy constraints (5.6) again is to show (C3) below.

$$\sum_{i\in\mathbf{I}_p}\frac{c_i\tau_i S_i(t+\tau_{i1})}{\delta_i}\geq\left(\sum_{i\in\mathbf{I}_p}c_i\tau_i\right)\cdot\left(\sum_{i\in\mathbf{I}_p}\tau_i\right),\forall p\in\mathbf{P}(\mathbf{I})\tag{C3}$$

Since task $i1$ is processed at time t, we know

$$S_{i1}(t+\tau_{i1})=S_{i1}(t)+\beta_{i1}-\tau_{i1}\delta_{i1}\tag{C4}$$

$$c_i\tau_i\frac{S_i(t+\tau_{i1})}{\delta_i}=c_i\tau_i\frac{S_i(t)}{\delta_i}-\tau_{i1}\cdot(c_i\tau_i),\quad\forall i\in\mathbf{I}\backslash\{i1\}\tag{C5}$$

First, we show that for any subset p including task $i1$, (C3) is satisfied. From constraints (5.2b) and (5.2c) of model MC, we know that

$$c_{i1}\frac{\beta_{i1}}{\delta_{i1}}\geq\sum_{i\in\mathbf{I}}c_i\tau_i$$

$$\Rightarrow c_{i1}\tau_{i1}\frac{S_{i1}(t+\tau_{i1})}{\delta_{i1}}=c_{i1}\tau_{i1}\frac{S_{i1}(t)+\beta_{i1}-\tau_{i1}\delta_{i1}}{\delta_{i1}}\geq c_{i1}\tau_{i1}\frac{S_{i1}(t)}{\delta_{i1}}+\tau_{i1}\cdot\sum_{i\in\mathbf{I}\backslash\{i1\}}c_i\tau_i\tag{C6}$$

Thus, by the summation of (C5) and (C6), (C3) of $\{p|i1\in\mathbf{I}_p\}$ can be deduced.

Second, by contradiction, we show that (C3) is also satisfied for any subset excluding task $i1$. Assuming (C3) is not satisfied for p such that $i1\notin\mathbf{I}_p$; i.e.,

$$\sum_{i\in\mathbf{I}_p}c_i\tau_i\frac{S_i(t+\tau_{i1})}{\delta_i}<(\sum_{i\in\mathbf{I}_p}c_i\tau_i)\cdot(\sum_{i\in\mathbf{I}_p}\tau_i)\tag{C7}$$

$$\Rightarrow\sum_{i\in\mathbf{I}_p}c_i\tau_i\frac{S_i(t)-\tau_{i1}\delta_i}{\delta_i}<(\sum_{i\in\mathbf{I}_p}c_i\tau_i)\cdot(\sum_{i\in\mathbf{I}_p}\tau_i)$$

$$\Rightarrow\sum_{i\in\mathbf{I}_p}c_i\tau_i\frac{S_i(t)}{\delta_i}<(\sum_{i\in\mathbf{I}_p}c_i\tau_i)\cdot(\sum_{i\in\mathbf{I}_p}\tau_i)+\tau_{i1}\sum_{i\in\mathbf{I}_p}c_i\tau_i\tag{C8}$$

The constraint of (C2) for p' such that $\mathbf{I}_{p'}=\mathbf{I}_p\cup\{i1\}$ is

$$c_{i1}\tau_{i1}\frac{S_{i1}(t)}{\delta_{i1}}+\sum_{i\in\mathbf{I}_p}c_i\tau_i\frac{S_i(t)}{\delta_i}\geq(c_{i1}\tau_{i1}+\sum_{i\in\mathbf{I}_p}c_i\tau_i)\cdot(\tau_{i1}+\sum_{i\in\mathbf{I}_p}\tau_i)\tag{C9}$$

From (C8) and (C9), we know

$$c_{i1}\tau_{i1}\frac{S_{i1}(t)}{\delta_{i1}} > c_{i1}\tau_{i1}\tau_{i1} + c_{i1}\tau_{i1}\sum_{i\in\mathbf{I}_p}\tau_i$$

$$\Rightarrow \frac{S_{i1}(t)}{\delta_{i1}} > \ \tau_{i1} + \sum_{i\in\mathbf{I}_p}\tau_i \tag{C10}$$

From (C1) and (C10), we know

$$\frac{S_i(t)}{\delta_i} \geq \frac{S_{i1}(t)}{\delta_{i1}} > \ \tau_{i1} + \sum_{i'\in\mathbf{I}_p}\tau_{i'} \quad \forall i \in \mathbf{I}_p$$

$$\Rightarrow \sum_{i\in\mathbf{I}_p} c_i\tau_i\frac{S_i(t+\tau_{i1})}{\delta_i} = \sum_{i\in\mathbf{I}_p} c_i\tau_i(\frac{S_i(t)}{\delta_i} - \tau_{i1}) > (\sum_{i\in\mathbf{I}_p} c_i\tau_i)\cdot(\sum_{i\in\mathbf{I}_p}\tau_i)$$

which contradicts (C7). Thus (C3) of $\{p|i1 \notin \mathbf{I}_p\}$ is also satisfied.

Thus, we have shown that after processing the task *i* with the smallest value of S_i/δ_i, inventory levels satisfy constraints (5.6) again. Therefore, there exists a feasible schedule for any horizon. Accordingly, we can construct a feasible solution of MF, as follows,

$$W_{it} = 1 \text{ if and only if } i \text{ starts at } t \text{ in this schedule} \quad \forall i, t$$

$$L_{it} = S_i - t\delta_i + \beta_i \sum_{t'<t-\tau_i} W_{it'} \quad \forall i, t$$

Therefore, when constraints (5.6) are satisfied, model MF is always feasible. ■

D. Proof of Proposition 5.6

Proposition 5.6: The projection of feasible region defined by constraints (5.8a) and (5.8b) on the subspace of $s = [S_1, S_2, \dots, S_{|I|}]^T$ *is the same as the feasible region defined by constraints (5.6).*

Proof: First we introduce two lemmas, which can be proved trivially.

Lemma D1: Assume the initial inventory levels s satisfy constraints (5.6). If two task subsets p_1 *and* p_2 *are disjoint (*$I_{p1} \cap I_{p2} = \emptyset$*), constraints (5.6) written for these two subsets cannot be both binding. Otherwise, the constraint (5.6) written for* p_3 *such that* $I_{p3} = I_{p1} \cup I_{p2}$ *will be violated.*

Lemma D2: Assume initial inventory levels s satisfy constraints (5.6). If two task subsets p_1 *and* p_2 *satisfy* $I_{p1} \backslash I_{p2} \neq \emptyset$ *and* $I_{p2} \backslash I_{p1} \neq \emptyset$*, constraints (5.6) written for these two subsets cannot be both binding. Otherwise, the constraints (5.6) written for* p_3, p_4 *such that* $I_{p3} = I_{p1} \cup I_{p2}, I_{p4} = I_{p1} \cap I_{p2}$ *cannot be satisfied at the same time.*

Let $\mathbf{S}^{(6)}$ ={$\boldsymbol{s}$|constraints (5.6)}, $\mathbf{S}^{(8)}$ ={$\boldsymbol{s}$|constraints (5.8a) and (5.8b)}. To show $\mathbf{S}^{(8)} = \mathbf{S}^{(6)}$, we show $\mathbf{S}^{(8)} \subseteq \mathbf{S}^{(6)}$ and $\mathbf{S}^{(6)} \subseteq \mathbf{S}^{(8)}$.

First, we show $\mathbf{S}^{(8)} \subseteq \mathbf{S}^{(6)}$. Consider any subset p. By multiplying (5.8a) by $c_i \tau_i$ on both side and adding together the inequalities of all $i \in \mathbf{I}_p$, we have

$$\sum_{i \in \mathbf{I}_p} c_i \tau_i \frac{S_i}{\delta_i} \geq \sum_{i \in \mathbf{I}_p} \left\{ c_i \tau_i \left[\tau_i + \sum_{i' < i} \tau_{i'} \left(\frac{c_{i'}}{c_i} + 1 - U_{i'i} \right) + \sum_{i' > i} \tau_{i'} \frac{c_{i'}}{c_i} U_{ii'} \right] \right\}$$

Rewriting the RHS, we have

$$\sum_{i \in \mathbf{I}_p} c_i \tau_i \frac{S_i}{\delta_i} \geq \sum_{i \in \mathbf{I}_p} c_i \tau_i \tau_i$$

$$+ \sum_{i \in \mathbf{I}_p} \left\{ c_i \tau_i \left[\sum_{i' \in \mathbf{I}_p : i' < i} \tau_{i'} \left(\frac{c_{i'}}{c_i} + 1 - U_{i'i} \right) + \sum_{i' \in \mathbf{I}_p : i' > i} \tau_{i'} \frac{c_{i'}}{c_i} U_{ii'} \right] \right\}$$

$$+\sum_{i\in\mathbf{I}_p}\left\{c_i\tau_i\left[\sum_{i'\notin\mathbf{I}_p:i'<i}\tau_{i'}\left(\frac{c_{i'}}{c_i}+1-U_{i'i}\right)+\sum_{i'\notin\mathbf{I}_p:i'>i}\tau_{i'}\frac{c_{i'}}{c_i}U_{ii'}\right]\right\}$$

From (5.8b), we know the last term on the RHS is non-negative. After simplifying the second sum term on the RHS, we have

$$\sum_{i\in\mathbf{I}_p}c_i\tau_i\frac{S_i}{\delta_i}\geq\sum_{i\in\mathbf{I}_p}c_i\tau_i\tau_i+\sum_{i\in\mathbf{I}_p,i'\in\mathbf{I}_p\setminus\{i\}}\{(c_i+c_{i'})\tau_i\tau_{i'}\}$$

Further rewriting the second sum term on the RHS, we have

$$\sum_{i\in\mathbf{I}_p}c_i\tau_i\frac{S_i}{\delta_i}\geq\sum_{i\in\mathbf{I}_p}c_i\tau_i\tau_i+\sum_{i\in\mathbf{I}_p}\left[c_i\tau_i\cdot\left(\sum_{i\in\mathbf{I}_p\setminus\{i\}}\tau_i\right)\right]$$

Then we can combine the two terms on the RHS

$$\sum_{i\in\mathbf{I}_p}c_i\tau_i\frac{S_i}{\delta_i}\geq(\sum_{i\in\mathbf{I}_p}c_i\tau_i)\cdot(\sum_{i\in\mathbf{I}_p}\tau_i)$$

Thus, (5.6) are derived. Thus, $\mathbf{S}^{(8)}\subseteq\mathbf{S}^{(6)}$.

Second, we show $\mathbf{S}^{(6)}\subseteq\mathbf{S}^{(8)}$. Since both $\mathbf{S}^{(6)}$ and $\mathbf{S}^{(8)}$ are convex, it suffices to show that each (finite) extreme point and infinite point of $\mathbf{S}^{(6)}$ belongs to $\mathbf{S}^{(8)}$.

Any of the extreme points, $[S_1,S_2,\ldots,S_{|\mathbf{I}|}]^{\mathrm{T}}$, of $\mathbf{S}^{(6)}$ is the intersection of at least $|\mathbf{I}|$ binding constraints of (5.6). From Lemmas D1 and D2, we know that if (5.6) of task subsets p and p' are both binding, then $\mathbf{I}_p\subseteq\mathbf{I}_{p'}$ or $\mathbf{I}_{p'}\subseteq\mathbf{I}_p$. Thus, the subsets of the $|\mathbf{I}|$ binding constraints follow the following format, if we order them in terms of the cardinality of the subset:

$$\begin{gathered}\mathbf{I}_{p1}=\{i1\}\\\mathbf{I}_{p2}=\{i1,i2\}\\\ldots\ldots\\\mathbf{I}_{p|\mathbf{I}|}=\{i1,i2,\ldots,i|\mathbf{I}|\}\end{gathered}\tag{D1}$$

Based on the binding constraints, we can find define $a_{ii'}$ accordingly as below,

$$\begin{cases} U_{im,in} = 0 & \text{if } \mathrm{ord}(im) < \mathrm{ord}(in) \\ U_{in,im} = 1 + \dfrac{c_{in}}{c_{im}} & \text{otherwise} \end{cases} \quad \forall\, m < n \tag{D2}$$

Note that the orders of m and n are related according to (D1); i.e., if $m < n, im \in \mathbf{I}_{pm}, im \in \mathbf{I}_{pn}$, $in \notin \mathbf{I}_{pm}, in \in \mathbf{I}_{pn}$. Also note that the orders of tasks are related according to (5.8b); i.e., $U_{im,in}$ is only defined for $\mathrm{ord}(im) < \mathrm{ord}(in)$. Obviously, , for $U_{ii'}$ defined in (D2), (5.8a) and (5.8b) are satisfied. Therefore any extreme point of $\mathbf{S}^{(6)}$ belongs to $\mathbf{S}^{(8)}$.

Now, consider any infinite point of $\mathbf{S}^{(6)}$. Assume that l dimensions are infinite, i.e., $S_{i1} = \cdots = S_{il} = +\infty$. We can define $U_{ii'}$ as follows, if $S_i = +\infty$

$$\begin{cases} U_{ii'} = 1 + \dfrac{c_i}{c_{i'}} & \text{if } i < i' \\ U_{i'i} = 0 & \text{otherwise} \end{cases} \quad \forall\, i: S_i = +\infty \tag{D3}$$

For the remaining $|\mathbf{I}| - l$ indices, we can find the extreme points for the subspace of $[S_{il+1}, S_{il+2}, \ldots, S_{i|\mathbf{I}|}]^{\mathrm{T}}$, and it would be the intersection of $|\mathbf{I}| - l$ binding constraints of (5.6). Thus, we can define $U_{ii'}$ following the same logic of (D2). Thus any infinite point of $\mathbf{S}^{(6)}$ also belongs to $\mathbf{S}^{(8)}$.

Therefore, we have shown that every extreme point and every infinite point of $\mathbf{S}^{(6)}$ belongs to $\mathbf{S}^{(8)}$, and $\mathbf{S}^{(6)} \subseteq \mathbf{S}^{(8)}$. ∎

E. Proof of Correctness of Constraints (SIIT)

Since the changeover constraints for different units are independent, we drop index j without loss of generality. Thus, constraints (SIIT) become

$$\sum_{t'=t-\tau_i+1}^{t} X_{it'} + \sum_{i' \neq i} \sum_{t'=t-\omega a_{ini'}}^{t-\omega b_{ini'}} X_{i't'} \leq 1 \quad \forall i, t, n \in \mathbf{N}_i \tag{E1}$$

Proof: To prove the correctness of (SIIT) for given (i, t), we need to show two things: (SIIT) forces all variables corresponding to $i' \neq i$ and $t' \in \mathbf{T}^C_{i'it}$ to be zero if $X_{it} = 1$, and no valid solutions are cut off.

First, we show that for any $i' \neq i$ and $t' \in \{t - \tau_{i'} - \sigma_{i'i} + 1, \dots, t - \tau_{i'}\}$, $X_{i't'}$ is included in at least one constraint of $n \in \mathbf{N}_i$, based on the following three observations. For all $i' \neq i$, we have:

(1) $X_{i't'}$ for $t' = t - \tau_{i'}$, which is the largest $t' \in \mathbf{T}^C_{i'it}$, is included in the constraint for $n = 1$ since $\omega b_{i1i'} \leq \tau_{i'} \leq \omega a_{i1i'}$ for either $i' \in \mathbf{IA}_{in}$ or $i' \in \mathbf{IB}_{in}$.

(2) $X_{i't'}$ for $t' = t - \tau_{i'} - \sigma_{i'i} + 1$, which is the first changeover time point, is included in the constraint for the smallest element in $\mathbf{N}_i$ for which $\mu_{in} \geq \sigma_{i'i}$ (denoted by $n_{i'}$).

(3) When n is less than $n_{i'}$, we have $\omega b_{i,n+1,i'} \leq \mu_{i,n+1} \leq \mu_{in} + \tau_{i'} = \omega a_{ini'} + 1$, which means that variables $X_{i't'}$ for all changeover times (i.e., $t' \in \mathbf{T}^C_{i'it}$) are included in at least one inequality.

Second, to prove that no valid solutions are cut off, we consider the following cases for given (i, t):

(1) The same task scheduled back-to-back. For i or i', the time indices of the binary variables included in one of the inequalities have a difference less than or equal to the processing time of this task, thus, no solution with the same task scheduled back-to-back is cut off.

(2) A task $i' \neq i$ scheduled after i. The smallest time index of the included $X_{it'}$ variables is $t^L_1 = t - \tau_i + 1$, and the largest time index of the included $X_{i't'}$ variables cannot be larger than $t^U_2 = t$. Since $t^U_2 - t^L_1 = \tau_i - 1 < \tau_i + \sigma_{ii'}$, no solution with i' scheduled after i is cut off.

(3) A task $i' \neq i$ is scheduled before i. The smallest time index of the included $X_{i't'}$ variables is $t_1^L = t - \tau_{i'} - \min\{\sigma_{i'i}, \mu_{in}\} + 1 \geq t - \tau_{i'} - \sigma_{i'i} + 1$, and the largest time index of the included $X_{it'}$ variables is $t_2^U = t$. Once again $t_2^U - t_1^L \leq \tau_{i'} + \sigma_{i'i} - 1$, so no solution with task i scheduled after i' is cut off.

(4) Two tasks, both in $\mathbf{IA}_{in}$, are scheduled back-to-back. The included variables have the same largest time index, $t - \mu_{in}$. Also, the difference between the largest and the smallest t' of the included $X_{i't'}$ variables, is $\tau_{i'} - 1$. Thus, no solution with these tasks scheduled one after another is cut off.

(5) Two tasks i_1, i_2, both in $\mathbf{IB}_{in}$, are scheduled back-to-back. The smallest time index of the included $X_{i_1t'}$ variables is $t_1^L = t - \tau_{i1} - \sigma_{i1,i} + 1$, while the time index of the included $X_{i_2t'}$ variables cannot be larger than $t_2^U = t - \sigma_{i1,i} + \sigma_{i1,i2}$. Since $t_2^U - t_1^L = \tau_{i1} + \sigma_{i1,i2} - 1$, no solution with task i_2 scheduled after i_1 is cut off.

(6) Task $i_1 \in \mathbf{IA}_{in}$ is scheduled before $i_2 \in \mathbf{IB}_{in}$. The smallest time index of the included $X_{i_1t'}$ variables is $t_1^L = t - \mu_{in} - \tau_{i1} + 1$, while the time index of $X_{i_2t'}$ variables cannot be larger than $t_2^U = t - \mu_{in} + \sigma_{i1,i2}$. Similar to (5), no solution with task i_1 scheduled before i_2 is cut off.

(7) Task $i_1 \in \mathbf{IA}_{in}$ is scheduled after $i_2 \in \mathbf{IB}_{in}$. The smallest time index of the included $X_{i_2t'}$ variables is $t_2^L = t - \tau_{i2} - \sigma_{i2,i} + 1$, while the time index of $X_{i_1t'}$ variables cannot be larger than $t_1^U = t - \mu_{in}$. The difference is $\tau_{i2} + \sigma_{i2,i} - \mu_{in} - 1$. Since $\mu_{in} > \sigma_{i2,i}$ ($i_2 \in \mathbf{IB}_{in}$), no solution with task i_1 scheduled after i_2 is cut off. ∎

F. Proof of Proposition 6.1

Proposition 6.1: Let $H = \{X \in \{0,1\}^{T\cdot|\mathbf{I}|\cdot|\mathbf{J}|}$*: subject to constraints (6.7) (6.23)}. Then each inequality in (6.23) is facet-defining for the convex hull of H, conv(H).*

Proof: Since changeovers in different units are independent, index j is dropped without loss of generality.Let $H = \{X \in \{0,1\}^{T\cdot|\mathbf{I}|}: \sum_i \sum_{t'=t-\tau_i+1}^{t} X_{it'} \leq 1, \forall t$ and

$\sum_{t'=t-\tau_i+1}^{t} X_{it'} + \sum_{i'\neq i} \sum_{t'=t-\omega a_{ini'}}^{t-\omega b_{ini'}} X_{i't'} \leq 1, \forall i, t, n \in \mathbf{N}_i\}$.

We want to prove that each face F_{itn} defined by (F1) is facet-defining for the convex hull, conv(H). We will prove it by showing that the face defined in expression (F1) has dimension $T\cdot|\mathbf{I}|-1$, which is one less than the dimension of conv(H), as there are $T\cdot|\mathbf{I}|$ X_{it}variables and H is full dimensional.

$$F_{itn} = \left\{ X_{it} \in \text{conv}(H): \sum_{t'=t-\tau_i+1}^{t} X_{it'} + \sum_{i'\neq i} \sum_{t'=t-\omega a_{ini'}}^{t-\omega b_{ini'}} X_{i't'} = 1 \right\} \quad \forall i, t, n \in \mathbf{N}_i \tag{F1}$$

To show that F_{itn} has dimension $T\cdot|\mathbf{I}|-1$, it suffices to show that it has $T\cdot|\mathbf{I}|$ affinely independent points.

For given (i, t, n), the following points belong to F_{itn}, and are affinely independent (one point is shown in each curly bracket, and for simplicity of presentation, we only list the non-zero variables):

(1) $\{X_{it'} = 1\}, t' \in \{t - \tau_i + 1, \dots, t\}$. These are τ_i points.

(2) $\{X_{it'} = 1, X_{i,t'+k\cdot\tau_i} = 1\}, t' \in \{1, \dots, t - \tau_i\} \cup \{t + 1, \dots, T\}, k \in \mathbb{Z}\backslash\{0\}$ and $t' + k \cdot \tau_i \in \{t - \tau_i + 1, \dots, t\}$. Note that k, as well as the k listed below, is negative if $t' > t$. These are $T - \tau_i$ points.

(3) For $i' \in \mathbf{IA}_{in}$, $\{X_{i't'} = 1\}, t' \in \{t - \omega a_{ini'}, \dots, t - \omega b_{ini'}\}$.

(4) For $i' \in \mathbf{IA}_{in}$, $\{X_{i't'} = 1, X_{i',t'+k\cdot\tau_i} = 1\}, t' \in \{1, \dots, t - \omega a_{ini'} - 1\} \cup \{t - \omega b_{ini'} + 1, \dots, T\}, k \in \mathbb{Z}\backslash\{0\}$ and $t' + k \cdot \tau_i \in \{t - \omega a_{ini'}, \dots, t - \omega b_{ini'}\}$. Cases (3) and (4) have $T \cdot |\mathbf{IA}_{in}|$ points.

(5) For $i' \in \mathbf{IB}_{in}$, $\{X_{i't'} = 1\}, t' \in \{t - \omega a_{ini'}, \dots, t - \omega b_{ini'}\}$.

(6) For $i' \in \mathbf{IB}_{in}$, $\{X_{i't'} = 1, X_{it} = 1\}, t' \in \{1, \dots, t - \omega a_{ini'} - 1\}$.

(7) For $i' \in \mathbf{IB}_{in}$ and $t' \in \{t - \omega b_{ini'} + 1, \dots, T\}$, based on the definition of $\omega b_{ini'}$, at least one of the following three scenarios is true, and we can select the point accordingly (in case when more than one scenario is true, we select one of them). Cases (5) – (7) have $T \cdot |\mathbf{IB}_{in}|$ points.

(7a) If $\omega b_{ini'} = \sigma_{i'i}$, select the point $\{X_{i't'} = 1, X_{i',t-\omega a_{ini'}} = 1\}$.

(7b) If $\omega b_{ini'} = \mu_{in} - \min_{i'' \in \mathbf{IA}_{in}} \sigma_{i''i'}$, select the point $\{X_{i't'} = 1, X_{i'',t-\omega a_{ini''}} = 1\}, i'' \in \mathbf{IA}_{in}$ and $\sigma_{i''i'} = \min_{i''' \in \mathbf{IA}_{in}} \sigma_{i'''i'}$.

(7c) If $\omega b_{ini'} = \max_{i'' \in \mathbf{IB}_{in} \setminus \{i'\}} \{\sigma_{i''i} - \sigma_{i''i'}\}$, select the point $\{X_{i't'} = 1, X_{i'',t-\omega a_{ini''}} = 1\}, i'' \in \mathbf{IB}_{in}$ and $\sigma_{i''i} - \sigma_{i''i'} = \max_{i''' \in \mathbf{IB}_{in} \setminus \{i'\}} \{\sigma_{i'''i} - \sigma_{i'''i'}\}$.

From the set definition, $|\mathbf{IA}_{in}| + |\mathbf{IB}_{in}| + 1 = |\mathbf{I}|$. Thus, there are totally $T \cdot |\mathbf{I}|$ points in (1) to (7). It is trivial to show the points are linearly independent, and thus affinely independent. ■

G. Algorithms

Algorithm 3.1. Dynamic network reduction (§3.2.1, 3.2.2).

1: **for** j
2: calculate parameters and subsets ζ_j^S, $\sigma_j^{\text{MIN}}, T_j, \mathbf{J}_j^R$;
3: $\mathbf{J}^T = \{j: \sigma_j^{\text{MIN}} > 0\}$;
4: **for** $j \in \mathbf{J}^T$
5: $\mathbf{J}^B = \mathbf{J}^B \cup \{j' \in \mathbf{J}_j^R \backslash \mathbf{J}^T : L0_{j'}^A - \int_0^{\eta+24T_{j'}} \rho_{j'}^T(t)dt < \zeta_{j'}^S\}$;
6: **if** $\left\{j' \in \mathbf{J}_j^R \backslash \mathbf{J}^T : L0_{j'}^A - \int_0^{\eta+24T_{j'}} \rho_{j'}^T(t)dt < \zeta_{j'}^S\right\} = \emptyset$ **then**
7: $\mathbf{J}^B = \mathbf{J}^B \cup \{j' \in \mathbf{J}_j^R \backslash \mathbf{J}^T : \sigma_{j'}^{\text{MAX}} = \max_{j'' \in \mathbf{J}_j^R} \sigma_{j''}^{\text{MAX}}\}$;
8: $\mathbf{J} = \mathbf{J}^T \cup \mathbf{J}^B \cup \{P\}$;
9: $\mathbf{A} = \mathbf{A} \setminus \{(j, j'): j \notin \mathbf{J} \text{ or } j' \notin \mathbf{J}\}$
10: **for** $(j, j') \in \mathbf{A}$
11: **if** (j, j') is not in the neighbor list or satisfies inequality (3.6) **then**
12: $\mathbf{A} = \mathbf{A} \setminus \{(j, j')\}$;
13: **if** $j \in \mathbf{J}^B$ and $j' \in \mathbf{J}^B$ and $\{j'' \in \mathbf{J}^T : j \in \mathbf{J}_{j''}^R \text{ and } j' \in \mathbf{J}_{j''}^R\} = \emptyset$ **then**
14: $\mathbf{A} = \mathbf{A} \setminus \{(j, j')\}$;

Note. Lines 1-8 preprocess customers; lines 9-14 preprocess network arcs.

Algorithm 3.2. Routes generation for VR subproblem (§3.3.1).

1: declare an array *cus*[];
2: **for** *u=1:cmax*
3: **for** *v=1:u*
4: **for** $j \in \mathbf{J}^C$ and different from *cus*[1], *cus*[2], ..., *cus*[*v*-1]
5: *cus*[*v*] = *j*;
6: **if** *v* = *u* **then**
7: $r = P \rightarrow cus[1] \rightarrow \cdots \rightarrow cus[u] \rightarrow P$;
8: calculate parameters as in equations (3.7)-(3.10);
9: **if** *r* satisfies all the criteria in §3.3.1 **then**
10: $\mathbf{R} = \mathbf{R} \cup \{r\}$;

Note. Lines 2-7 list all possible routes; line 8 calculates the corresponding parameters; lines 9-10 verify the condition whether a route should be included in set R.

Algorithm 3.3. Set definition for heuristic integer cut generation (§3.5.3).

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

Note. Line 1 checks if a truck is assigned to some routes that lead to infeasibility; lines 2-7 update the sets denoting infeasible route combinations using OptE or OptR.

Algorithm 3.4. Parameter updating for the VR subproblem (§3.5.4).

1: **for** i, r, l: $l \in \mathbf{L}^{R}_{r} \backslash \mathbf{L}^{2}$
2: **if** $\tau^{R}_{r} < \sum_{n,k,n'} X_{i,n,k,n',l}(E^{I}_{i,n} - S^{I}_{i,n})$
3: $\tau x_{i,r} = \sum_{n,k,n'} X_{i,n,k,n',l}(E^{I}_{i,n} - S^{I}_{i,n}) - \tau^{R}_{r}$

Algorithm 4.1. Check long-term model vessel availability.

1: *resolve* = no;
2: *nmax* = 0;
3: **for** $\lambda^{LA} \le t \le \lambda^{LB}$, $i \in \mathbf{I} \backslash \mathbf{I}^{R}$: $\sum_{j} W^{L}_{i,vc,jt} = 1$
4: *nmax* = *nmax* +1;
5: $\varepsilon_L(t) = \varepsilon_L(t_{LA}) + [\varepsilon_L(t_{LB}) - \varepsilon_L(t_{LA})] \cdot \frac{t - t_{LA}}{t_{LB} - t_{LA}}$;
6: sample *random* from a continuous uniform distribution of interval [0,1];
7: **if** $random \le \varepsilon_L(t)$ **then**
8: $\delta^{L+}_{nmax} = t$;
9: **else**
10: $\delta^{L+}_{namx} = \text{ceil}(random \cdot \lambda^{LC})$;
11: *resolve* = yes;
12: break;

Note. If all desired vessels in long-term mode are available, the algorithm ends with resolve equal to no; otherwise, resolve returns yes. In line 5, $\varepsilon_L(t)$ denotes the probability that a vessel is available at time t. In line 10, λ^{LC} denotes the earliest time a vessel in long-term mode is guaranteed to be available. In the case study, $t_{LA} = 7, t_{LB} = 14, \varepsilon_L(t_{LA}) = 0.85, \varepsilon_L(t_{LA}) = 1, \lambda^{LC} = 14$.

Algorithm 4.2. Update availability of vessels in short-term mode.

1: **for** l
2: **for** $n \leq nlast_l$
3: $\delta_{ln}^{S+} = null;$
4: **for** n: $\delta_{ln}^{S+} \neq null$
5: sample $random1$ from a continuous uniform distribution of interval [0,1];
6: **if** $random1 \in [0, \varepsilon_{SA}]$ **then**
7: $\delta_{ln}^{S+} = \delta_{ln}^{S-} - 1;$
8: **else if** $random1 \in [\varepsilon_{SA}, \varepsilon_{SA} + \varepsilon_{SB}]$ **then**
9: $\delta_{ln}^{S+} = \delta_{ln}^{S-} - 2;$
10: **else if** $random1 \in [\varepsilon_{SA} + \varepsilon_{SB}, \varepsilon_{SA} + \varepsilon_{SB} + \varepsilon_{SC}]$ **then**
11: $\delta_{ln}^{S+} = \delta_{ln}^{S-};$
12: **else if** $random1 \in [\varepsilon_{SA} + \varepsilon_{SB} + \varepsilon_{SC}, \varepsilon_{SA} + \varepsilon_{SB} + \varepsilon_{SC} + \varepsilon_{SD}]$ **then**
13: $\delta_{ln}^{S+} = \delta_{ln}^{S-} + 1;$
14: **else**
15: $\delta_{ln}^{S+} = null;$
16: sort the values of δ_{ln}^{S+} over index *n* in ascending order so that $\delta_{ln}^{S+} \leq \delta_{l,n+1}^{S+}$ and all the *null* values are moved to the larger end of *n*;
17: find the smallest index *n'* such that $\delta_{ln'}^{S+} = null;$
18: sample *random*2 from a discrete uniform distribution of {*newA*, *newA*+1,…, *newB*};
19: $\delta_{ln'}^{S+} = random2;$

Note. The availability profiles of vessels rented in short-term mode are cluster-specific. For each cluster l, we (1) remove the vessels that were reserved in the previous period (lines 2-3); (2) update the times when vessels become available (lines 4-15); (3) sort those times in ascending order (line 16); and (4) generate new availability profile (lines 17-19). In line 2, $nlast_l$ is the number of trips in cluster l reserved in the last period. Parameters $\varepsilon_{SA}, \varepsilon_{SB}, \varepsilon_{SC}, \varepsilon_{SD}$ denote the probability that the time a vessel becomes available remains unchanged, decreases by 1 period, increases by 1 period, and increases by 2 periods, respectively. For example, in the case that the time a vessel becomes available is unchanged, the new δ_{ln}^{S} is the old δ_{ln}^{S} minus one as in line 7, because the horizon has been rolled forward by one day. It is also possible that a previously available vessel is reserved by another party, and thus becomes unavailable, as shown in line 15. In the case study, $\varepsilon_{SA} = 0.75, \varepsilon_{SB} = 0.05, \varepsilon_{SC} = 0.05, \varepsilon_{SD} = 0.05, newA = 9, newB = 12$.

Algorithm 4.3. Check availability of vessels in short-term mode.

for l
ncurrent = 0;
for $\lambda^{SA} \leq t \leq \lambda^{SB}$
for $(j, j') \in \mathbf{A}_l \backslash \mathbf{A}_t^R : W_{jj't}^S = 1$
ncurrent = *ncurrent* +1;
if $t < \delta_{l,ncurrent}^{S+}$ **then**
resolve = yes;
break;

Note. The availability of vessels rented in short-term mode is checked for each cluster; if some desired vessels are not available, resolve returns yes.

Algorithm 4.4. Update parameters due to trip delays.

for $i, j, t > 1$: $\widehat{W}^{L-}_{i,vc,jt} = 1$
sample *random* from a continuous uniform distribution of interval [0,1];
if $random \in [0, \varepsilon_{DI1}]$ **then**
$\widehat{W}^{L+}_{i,vc,j,t+1} = 1$;$\widehat{W}^{L+}_{i,vc,jt}$ =0;
else if $random \in [\varepsilon_{DI1}, \varepsilon_{DI1} + \varepsilon_{DI2}]$ **then**
$\widehat{W}^{L+}_{i,vc,j,t+2} = 1$; $\widehat{W}^{L+}_{i,vc,jt}$ =0;
for $j, j', t > 1$: $\widehat{W}^{S-}_{jj't} = 1$
sample *random* from a continuous uniform distribution of interval [0,1];
if $random \in [0, \varepsilon_{DI1}]$ **then**
$\widehat{W}^{S+}_{jj',t+1} = 1$; $\widehat{W}^{S+}_{jj't}$ =0;
else if $random \in [\varepsilon_{DI1}, \varepsilon_{DI1} + \varepsilon_{DI2}]$**then**
$\widehat{W}^{S+}_{jj',t+2} = 1$; $\widehat{W}^{S+}_{jj't}$ =0;
for $i, j, t > 1$: $\hat{X}^{L-}_{ijt} = 1$
sample *random* from a continuous uniform distribution of interval [0,1];
if $random \in [0, \varepsilon_{DO1}]$ **then**
$\hat{F}^{L+}_{ijm,t+1} = \hat{F}^{L-}_{ijmt}$; $\hat{F}^{L+}_{ijmt} = 0$; $\hat{X}^{L+}_{ij,t+1} = 1$; $\hat{X}^{L+}_{ijt} = 0$;
else if $random \in [\varepsilon_{DO1}, \varepsilon_{DO1} + \varepsilon_{DO2}]$**then**
$\hat{F}^{L+}_{ijm,t+2} = \hat{F}^{L-}_{ijmt}$; $\hat{F}^{L+}_{ijmt} = 0$; $\hat{X}^{L+}_{ij,t+2} = 1$; $\hat{X}^{L+}_{ijt} = 0$;
for $j, j', t > 1$: $\sum_m \hat{F}^{S-}_{jj'mt} > 0$
sample *random* from a continuous uniform distribution of interval [0,1];
if $random \in [0, \varepsilon_{DO1}]$ **then**
$\hat{F}^{S+}_{jj'm,t+1} = \hat{F}^{S-}_{jj'mt}$; $\hat{F}^{S+}_{jj'mt} = 0$;
else if $random \in [\varepsilon_{DO1}, \varepsilon_{DO1} + \varepsilon_{DO2}]$**then**
$\hat{F}^{S+}_{jj'm,t+2} = \hat{F}^{S-}_{jj'mt}$; $\hat{F}^{S+}_{jj'mt} = 0$;
for $i, j \neq vc, j'$: $W^{L}_{ijj'1} = 1$
sample *random* from a continuous uniform distribution of interval [0,1];
if $random \in [0, \varepsilon_{DT1}]$ **then**
$\hat{F}^{L+}_{ij'm,\tau_{jj'}+2} = F^{L}_{ijj'm1}$; $\hat{X}^{L+}_{ij',\tau_{jj'}+2} = 1$;
else if $random \in [\varepsilon_{DT1}, \varepsilon_{DT1} + \varepsilon_{DT2}]$**then**
$\hat{F}^{L+}_{ij'm,\tau_{jj'}+3} = F^{L}_{ijj'm1}$; $\hat{X}^{L+}_{ij',\tau_{jj'}+3} = 1$;
else
$\hat{F}^{L+}_{ij'm,\tau_{jj'}+1} = F^{L}_{ijj'm1}$; $\hat{X}^{L+}_{ij',\tau_{jj'}+1} = 1$;
for j, j': $W^{S}_{jj'1} = 1$
sample *random* from a continuous uniform distribution of interval [0,1];
if $random \in [0, \varepsilon_{DT1}]$ **then**
$\hat{F}^{S+}_{jj'm,\tau_{jj'}+2} = F^{S}_{jj'm1}$;
else if $random \in [\varepsilon_{DT1}, \varepsilon_{DT1} + \varepsilon_{DT2}]$**then**
$\hat{F}^{S+}_{jj'm,\tau_{jj'}+3} = F^{S}_{jj'm1}$;
else

40: $\hat{F}^{S+}_{jj'm,\tau_{jj'}+1} = F^{S}_{jj'm1}$;

Note. There are three types of trip delays: (1) a reserved vessel that is yet to arrive can be late for 1 period with a probability of ε_{DI1} or 2 periods with a probability ε_{DI2} (lines 1-6 and 7-12 respectively for long- and short-term mode); (2) an on-going trip can have a delay of 1/2 periods with a probability $\varepsilon_{DO1}/\varepsilon_{DO2}$ (lines 13-18 and 19-24 for long- and short-term mode, respectively); and (3) a pick-up/delivery can be 1/2 periods longer with a probability $\varepsilon_{DT1}/\varepsilon_{DT2}$ (lines 25-32,33-40 respectively for long- and short-term mode). Note that even though the probability of delay at each time period does not depend on trip length, longer trips tend to have larger delays, because they have more time periods during which delays can be observed. In the case study, $\varepsilon_{DI1} = 0.15, \varepsilon_{DI2} = 0.05, \varepsilon_{DO1} = 0.08, \varepsilon_{DO2} = 0.02, \varepsilon_{DT1} = 0.15, \varepsilon_{DT2} = 0.05$.

Algorithm 4.5. Update pick-up windows.

for $j \in \mathbf{J}^{TP}, k \in \mathbf{K}_j$: $\sigma^{OS-}_{jk} = \lambda^{PU}$ and $\sigma^{OE-}_{jk} - \sigma^{OS-}_{jk} = 10$
    sample *random* from a discrete uniform distribution of {-*newC*, -*newC*+1,…, *newC*};
    $\sigma^{OS+}_{jk} = \sigma^{OS-}_{jk} + random1$;
    sample *random* from a discrete uniform distribution of {*newD*, *newD*+1,…, *newE*};
    $\sigma^{OE+}_{jk} = \sigma^{OS+}_{jk} + random2$;
    update θ_{jkt} according to equation (4.22);

Note. For pick-up windows whose estimated start time is $t = \lambda^{PU}$, the actual start time and window length are specified. Parameter newC is the maximum adjustment of the start time; newD/newE is the shortest/longest window length. In the case study, $newC = 3, newD = 2, newE = 3$.

Algorithm 4.6. Update initial inventory.

for $j \in \mathbf{J}^{OP} \cup \mathbf{J}^{C}$
    sample *random* from a normal distribution $\mathcal{N}(0, \sigma_{AF})$;
    $\hat{L}^{+}_{jm0} = L_{jm1} + random \cdot \rho^{-}_{jm1}$;

Note. The real-time initial inventory level is updated using a normal distribution. The consumption/production rate in the last period is included in L_{jm1}. In the case study, $\sigma_{AF} = 0.05$.

Algorithm 4.7. Update forecast consumption/production rate.

for $j \in \mathbf{J}^{OP} \cup \mathbf{J}^{C}, t > 1$
    sample *random* from a normal distribution $\mathcal{N}(0, \sigma_{FF})$;
    $\rho^{+}_{jmt} = (1 + random) \cdot \rho^{-}_{jmt}$;

Note. Consumption/production forecast rate are updated using a normal distribution. In the case study, $\sigma_{FF} = 0.05$.

Algorithm 4.8. Fix variables according to decisions made previously.

1: **for** $i, j, t < \lambda^{LR}$

2: fix $W^{L}_{ij,vc,t}$ to $\widehat{W}^{L+}_{ij,vc,t}$;

3: **for** $i, j, t < \delta^{LE}$;

4: fix $W^{L}_{i.vc.jt}$ to $\widehat{W}^{L+}_{i,vc,jt}$;

5: **for** $j, j', t < \lambda^{SA}$

6: fix $W^{S}_{jj't}$ to $\widehat{W}^{S+}_{jj't}$;

7: **for** $j, j', \lambda^{SA} \leq t \leq \lambda^{SB}: \widehat{W}^{S+}_{jj't} = 1$

8: fix $W^{S}_{jj't}$ to $\widehat{W}^{S+}_{jj't}$;

Note. Variables related to three types of decisions are fixed. First, the vessel company should be notified λ^{LR} periods before returning a long-term rental (lines 1-2). Second, long-term renting decisions that were made previously are fixed (lines 3-4); δ^{LE} is the earliest time when a vessel in long-term mode becomes available (updated in lines 20-23 in Algorithm 4.1). Finally, short-term renting decisions are fixed (lines 5-8). In the case study, $\lambda^{LR} = 15$.

Algorithm 4.9. The procedure to roll the horizon one step forward.

1: **if** *resolve* = no **then**

2: **for** $i, j, \lambda^{LA} \leq t \leq \lambda^{LB}: W^{L}_{i,vc,jt} = 1$

3: $\widehat{W}^{L+}_{i,vc,jt} = W^{L}_{i,vc,jt}$;

4: **else**

5: **for** $i, j, \lambda^{LA} \leq t \leq \delta^{L}_{namx}: W^{L}_{i,vc,jt} = 1$

6: $\widehat{W}^{L+}_{i,vc,jt} = W^{L}_{i,vc,jt}$;

7: **for** $j, j', \lambda^{SA} \leq t \leq \lambda^{SB}: W^{S}_{jj't} = 1$

8: $\widehat{W}^{S+}_{jj't} = W^{S}_{jj't}$;

9: **for** i, j

10: $\widehat{W}^{L+}_{ij,vc,\lambda^{LR}} = W^{L}_{ij,vc,\lambda^{LR}}$;

11: $\mathbf{I}^{R} = \left\{ i \middle| \sum_{j,t \leq \lambda^{LB}} \widehat{W}^{L+}_{i,vc,jt} = 1 \right\}$;

12: **for** $t \leq \lambda^{SB}$

13: $\mathbf{A}^{R}_{t} = \left\{ (j, j') \middle| \widehat{W}^{S+}_{jj't} = 1 \right\}$;

14: $C_1(d) = C^{MH}_1 + C^{OF}_1 + C^{UF}_1 + C^{FT}_1 + C^{VT}_1 + C^{FL}_1 + C^{EL}_1 + C^{S}_1$;

15: update parameters $\widehat{W}^{L+}_{ijj't}, \widehat{W}^{S+}_{jj't}, \hat{X}^{L+}_{ijt}, \hat{F}^{L+}_{ijmt}, \hat{F}^{S+}_{j'jmt}$ due to trip delays by Algorithm 4.4;

16: update pick-up windows $\sigma^{OS+}_{jk}, \sigma^{OE+}_{jk}, \theta^{+}_{jkt}$ by Algorithm 4.5;

17: update initial inventory levels $\hat{L}^{+}_{jm0}$ by Algorithm 4.6;

18: update forecast consumption/production rate ρ^{+}_{jmt} by Algorithm 4.7;

19: calculate how many periods vessel *i* has been rented for, and update χ_{it};

20: **if** *resolve* = no **then**

21: $\delta^{LE+} = \max(\delta^{LE-} - 1, \lambda^{LA})$;

22: **else**

23: $\delta^{LE+} = \max(\delta^{L}_{namx} - 1, \lambda^{LA})$.

24: roll horizon one period forward by modifying index *t* for all related parameters (lines 25-26);

25: **for** *i*, *j*, *j'*, *t*

26: $\widehat{W}^{L+}_{ijj't} = \widehat{W}^{L+}_{ijj't+1}$ (Similarly for $\widehat{W}^{S+}_{jj't}, \hat{X}^{L+}_{ijt}, \hat{F}^{L+}_{ijmt}, \hat{F}^{S+}_{jj'mt}, \theta^{+}_{jkt}, \rho^{+}_{jmt}$);

27: get the new information of consumption/production rate and orders at $t = \eta$;

28: $d = d + 1;$

Note. In the algorithm, (1) the new long-/short-term renting and long-term returning decisions are updated (lines 1-10); (2) the reserved vessels/trips are updated (lines 11-13); (3) the implementation cost for the current period is recorded (line 14); and (4) all parameters (including the stochastic parameters) are updated (lines 15-28).

Algorithm 5.1. Algorithm to obtain region $\mathbf{S}^{\mathrm{F}}$.

for $m \in \mathbf{M}$
    define min_m, max_m;
    $S_m = min_m$;
$endFlag = 0; changeFlag = 0; \mathbf{S}^{\mathrm{F}} = \emptyset$;
while $endFlag = 0$
    if $[S_{m1}, S_{m2}, \dots, S_{m|\mathbf{M}|}]^{\mathrm{T}} \notin \mathbf{S}^{\mathrm{F}}$
        run model MF;
        if model MF is feasible
            $\mathbf{S}^{\mathrm{F}} = \mathbf{S}^{\mathrm{F}} \cup \left\{ \left[S'_{m1}, S'_{m2}, \dots, S'_{m|\mathbf{M}|} \right]^{\mathrm{T}} \middle| S'_m \geq S_m, \forall m \right\}$
    for $m \in \mathbf{M}$: $changeFlag = 0$
        if $S_m < max_m$
            $S_m = S_m + \varepsilon$;
            $changeFlag = 1$;
        else
            $S_m = min_m$;
    if $changeFlag = 0$
        $endFlag = 1$;
    $changeFlag = 0$;

Note. min_m, max_m *denote the range of* S_m; ε *denotes the resolution of the discretization.*

Algorithm 6.1. Procedure to generate the parameters and sets used in (SIIT).

for $j, i \in \mathbf{I}_j$
    $n = 1;\ \mu_{ij1} = \min_{i' \neq i} \tau_{i'j}$; $stopflag = 0$;
    while $stopflag = 0$
        $\mathbf{IA}_{ijn} = \left\{ i' \middle| i' \neq i, \mu_{ijn} \leq \sigma_{i'ij} \right\}$; $\mathbf{IB}_{ijn} = \{ i' | i' \neq i, \mu_{ijn} > \sigma_{i'ij} \}$;
        for $i' \in \mathbf{IA}_{ijn}$
            $\omega a_{ijni'} = \tau_{i'j} + \mu_{ijn} - 1;\ \omega b_{ijni'} = \mu_{ijn}$;
        for $i' \in \mathbf{IB}_{ijn}$
            $\omega a_{ijni'} = \tau_{i'j} + \sigma_{i'ij} - 1$;
            $\omega b_{ijni'} = \max\{ \sigma_{i'ij}, \mu_{ijn} - \min_{i'' \in \mathbf{IA}_{ijn}} \sigma_{i''i'j}, \max_{i'' \in \mathbf{IB}_{ijn} \setminus \{i'\}} (\sigma_{i''ij} - \sigma_{i''i'j}) \}$;
        if $\mu_{ijn} \geq \max_{i' \neq i} \sigma_{i'ij}$
            $stopflag = 1$; set $\mathbf{N}_{ij} = \{1, \dots, n\}$;
        else
            $\mu_{ij,n+1} = \min \left\{ \mu_{ijn} + \min_{i' : \mu_{ijn} < \sigma_{i'ij}} \tau_{i'j}, \max_{i' \neq i} \sigma_{i'ij} \right\};\ n = n + 1$;

Bibliography

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

Agra, A.; Christiansen, M.; Delgado, A. Discrete time and continuous time formulations for a short sea inventory routing problem *Optim. Eng.* **2016**, doi:10.1007/s11081-016-9319-0.

Ahujia, R.K.; Magnanti, T.L.; Orlin, J.B. *Network flows: theory, algorithms, and applications*; Prentice Hall: Eaglewood Cliffs, NJ, 1993.

Al-Ameri, T.A.; Shah, N.; Papageorgiou, L.G. Optimization of vendor-managed inventory systems in a rolling horizon framework. *Comput. Ind. Eng.* **2008**, *54(4)*, 1019-1047.

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

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

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

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

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

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

Baker, K.R. An analysis of terminal conditions in rolling schedules. *Eur. J. Oper. Res.* **1981**, *7*(4):355-361.

Balasubramanian, J.; Grossmann, I.E. Scheduling optimization under uncertainty - an alternative approach. *Comput. Chem. Eng.* **2003**, *27*(4), 469-490.

Balasubramanian, J.; Grossmann, I.E. Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty. *Ind. Eng. Chem. Res.* **2004**, *43(14)*, 3695-3713.

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

Bassett, M.H.; Pekny, J.F.; Reklaitis, G.V. Using detailed scheduling to obtain realistic operating policies for a batch processing facility. *Ind. Eng. Chem. Res.* **1997**, *36(5)*, 1717-1726.

Bell, W.J.; Dalberto, L.M.; Fisher, M.L.; Greenfield, A.J.; Jaikumar, R.; Kedia, P.; Mack, R.G.; Prutzman, P.J. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, **1983**, *13*(6), 4-23.

Bonfill, A.; Bagajewicz, M.; Espuña, A.; Puigjaner, L. Risk management in the scheduling of batch plants under uncertain market demand. *Ind. Eng. Chem. Res.* **2004**, *43*(3), 741-750.

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

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

Coelho, L.C.; Cordeau, J.F.; Laporte, G. Consistency in multi-vehicle inventory-routing*. Transport. Res. C-EMER.* **2012**, *24*, 270-287.

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

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

Cui, J.; Engell, S. Medium-term planning of a multiproduct batch plant under evolving multi-period multi-uncertainty by means of a moving horizon strategy. *Comput. Chem. Eng.* **2010**, *34(5)*, 598-619.

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

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

Dondo, R.; Mendez, C.A.; Cerda; J. Optimal management of logistic activities in multi-site environments. *Comput. Chem. Eng.* **2008**, *32* (11), 2547-2569.

Dondo, R.; Mendez, C.A.; Cerda; J. Managing distribution in supply chain networks. *Ind. Eng. Chem. Res.* **2009**, *48* (22), 9961-9978.

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

Dong, Y.; Maravelias, C.T.; Pinto, J.M.; Sundaramoorthy, A. Solution Methods for Vehicle-based Inventory Routing Problems. *Comput. Chem. Eng.* **2017**, doi: 10.1016/j.compchemeng.2017.02.036.

Dong, Y.; Maravelias, C.T.; Jerome, N.F. Reoptimization Framework and Policy Analysis for Maritime Inventory Routing under Uncertainty. *Submitted.*

Dong, Y.; Maravelias, C.T. Terminal Constraints on Inventory Levels for Online Scheduling. *In preparation.*

Drexl A.; Kimms A. Lot sizing and scheduling—survey and extensions. *Eur. J. Oper. Res.* **1997,** *99*(2):221-235.

Engineer, F.G.; Furman, K.C.; Nemhauser, G.L.; Savelsbergh, M.W.; Song, J.H. A branch-price-and-cut algorithm for single–product maritime inventory routing. *Oper. Res.* **2012**, *60* (1), 106-122.

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

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

Glankwamdee, W.; Linderoth, J.; Shen J.; Connard, P.; Hutton, J. Combining optimization and simulation for strategic and operational industrial gas production and distribution. *Comput. Chem. Eng.* **2008**, *32* (11), 2536-2546.

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

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

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

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

Gupta, D.; Maravelias, C.T. On deterministic online scheduling: major considerations, paradoxes and remedies. *Comput. Chem. Eng.* **2016**, *94*, 312-330.

Gupta, D.; Maravelias, C.T.; Wassick, J.M. From rescheduling to online scheduling. *Chem. Eng. Res. Des.* **2016**, *116*, 83-97.

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

Harjunkoski, I.; Maravelias, C.T.; Bongers, P.; Castro, P.M.; Engell, S.; Grossmann, I.E.; Hooker, J.; Méndez, C.;Sand, G.; Wassick, J. Scope for industrial applications of production scheduling models and solution methods. *Comput. Chem. Eng.* **2014**, *62*, 161-193.

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

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

Janak, S.L.; Lin, X.; Floudas, C.A. A new robust optimization approach for scheduling under uncertainty: II. Uncertainty with known probability distribution. *Comput. Chem. Eng.* **2007**, *31(3)*, 171-195.

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

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

Karmarkar, U.S.; Schrage, L. The Deterministic Dynamic Product Cycling Problem. Oper. Res. **1985**, *33*(2): 326-345.

Kelly, J.D.; Zyngier, D. An Improved MILP Modeling of Sequence-Dependent Switchovers for Discrete-Time Scheduling Problems. Ind. Eng. Chem. Res. **2007**, *46*(14): 4964-4973.

Kleywegt, A.J.; Nori, V.S.; Savelsbergh, M.W.P. The stochastic inventory routing problem with direct deliveries. *Transport. Sci.* **2002**, *36(1)*, 94-118.

Kondili, E.; Pantelides, C.C.; Sargent, R.W.H. A General Algorithm for Short-Term Scheduling of Batch-Operations. 1. Milp Formulation. *Comput. Chem. Eng.* **1993**, *17*: 211-227.

Kreipl, S.; Pinedo, M. Planning and scheduling in supply chains: an overview of issues in practice. *Prod. Oper. Manag.* **2004**, *13*(1), 77-92.

Laporte, G.; Nobert, Y; Taillefer, S. Solving a family of multi-depot vehicle routing and location-routing problems. *Transport Sci.* **1988**, *22*(3):161-172.

Laporte, G. Fifty years of vehicle routing. *Transport. Sci.* **2009**, *43* (4), 408-416.

Lappas, N.H.; Gounaris, C.E. Multi-stage adjustable robust optimization for process scheduling under uncertainty. *AIChE J.* **2016**, *62*(5), 1646-1667.

Lei, L.; Liu, S.; Ruszczynski, A.; Park, S. On the integrated production, inventory, and distribution routing problem. *IIE Trans.* **2006**, *38* (11), 955-970.

Li, Z.; Ierapetritou. Process scheduling under uncertainty: Review and challenges. *Comput. Chem. Eng.* **2008a**, *32*, 715-727.

Li, Z; Ierapetritou, M.G. Robust optimization for process scheduling under uncertainty. *Ind. Eng. Chem. Res.* **2008b**, *47*(12), 4148-4157.

Lima, R.M.; Grossmann, I.E.; Jiao, Y. Long-term scheduling of a single-unit multi-product continuous process to manufacture high performance glass. *Comput. Chem. Eng.* **2011**, *35*(3), 554-574.

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

Maravelias, C.T. On the combinatorial structure of discrete-time MIP formulations for chemical production scheduling. *Comput. Chem. Eng.* **2012a**, *38*, 204-212.

Maravelias, C.T. General framework and modeling approach classification for chemical production scheduling. *AIChE J.* **2012b**, *58*(6), 1812-1828.

Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O.M. Constrained model predictive control: stability and optimality. *Automatica .***2000**, *36*(6), 789-814.

Méndez, C.A.; Cerdá, J.; Grossmann, I.E.; Harjunkoski, I; Fahl, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng.* **2006**, *30(6)*, 913-946.

Nagy, G.; Salhi, S. Location-routing: Issues, models and methods. *Eur. J. Oper. Res.* **2007**, *177* (2), 649-672.

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

Moniz, S.; Barbosa-Póvoa, A.P.; de Sousa, J.P. New General Discrete-Time Scheduling Model for Multipurpose Batch Plants. *Ind. Eng. Chem. Res.* **2013**, *52*(48): 17206-17220.

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

Novas, J.M.; Henning, G.P. Reactive scheduling framework based on domain knowledge and constraint programming. *Comput. Chem. Eng.* **2010**, *34(12)*, 2129-2148.

Ouelhadj, D.; Petrovic, S. A survey of dynamic scheduling in manufacturing systems. *J. Sched.* **2009**, *12*, 417-431.

Papageorgiou, D.J.; Keha, A.B.; Nemhauser, G.L.; Sokol, J. Two-stage decomposition algorithms for single product maritime inventory routing. *INFORMS J. Comput.* **2014a**, *26*(4), 825-847.

Papageorgiou, D.J.; Nemhauser, G.L.; Sokol, J.; Cheon, M.S.; Keha, A.B. MIRPLib - A library of maritime inventory routing problem instances: Survey, core model, and benchmark results. *Eur. J. Oper. Res.* **2014b**, *235*(2), 350-366.

Persson, J.A.; Göthe-Lundgren, M. Shipment planning at oil refineries using column generation and valid inequalities. *Eur. J. Oper. Res.* **2005**, *163*(3), 631-652.

Proth, J.M. Scheduling: New trends in industrial environment. *Annu. Rev. Control.* **2007**, *31*(1):157-166.

Raa, B. Fleet optimization for cyclic inventory routing problems. *Int. J. Prod. Econ.* **2015**, *160*, 172-181.

Rancourt, M.E.; Cordeau, J.F.; Laporte, G. Long-haul vehicle routing and scheduling with working hour rules. *Transport. Sci.* **2013**, *47*(1), 81-107.

Ronen, D. Marine inventory routing: shipments planning. *J. Oper. Res. Soc.* **2002**, *53(1)*, 108-114.

Sahin, F.; Robinson, E.P.; Gao, L.-L. Master production scheduling policy and rolling schedules in a two-stage make-to-order supply chain. *Int. J. Prod. Econ.* **2008**, *115(2)*, 528-541.

Sahinidis, N.V.; Grossmann, I.E. MINLP Model for Cyclic Multiproduct Scheduling on Continuous Parallel Lines. *Comput. Chem. Eng.* **1991**, *15*(2): 85-103.

Sana, S.S.; Goyal, S.K. (Q,r,L) model for stochastic demand with lead-time dependent partial backlogging. *Ann. Oper. Res.* **2015**, *233*(1), 401-410.

Sand, G.; Engell, S.; Märkert, A., Schultz, R.; Schulz, C. Approximation of an ideal online scheduler for a multiproduct batch plant. *Comput. Chem. Eng.* **2000**, *24*(2-7), 361-367.

Savelsbergh, M.; Song, J.H. Inventory routing with continuous moves. *Comput. Oper. Res.* **2007**, *34*(6), 1744-1763.

Shah, N.; Pantelides, C.C.; Sargent, R.W.H. Optimal periodic scheduling of multipurpose batch plants. *Ann. Oper. Res.* **1993**, *42*(1), 193-228.

Stadtler, H. Improved rolling schedules for the dynamic single-level lot-sizing problem. *Manag. Sci.* **2000**, *46*(2):318-326.

Shen, Z.J.M.; Qi, L. Incorporating inventory and routing costs in strategic location models. *Eur. J. Oper. Res.* **2007**, *179* (2), 372-389.

Shen, Q.; Chu, F.; Chen, H. A Lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. *Comput. Chem. Eng.* **2011**, *35*(10), 2113-2123.

Singh, T.; Arbogast, J.E.; Neagu, N. An incremental approach using local-search heuristic for inventory routing problem in industrial gases. *Comput. Chem. Eng*. **2015**, *80*, 199-210.

Siswanto, N.; Essam, D.; Sarker, R. Solving the ship inventory routing and scheduling problem with undedicated compartments. *Comput. Ind. Eng.* **2011**, *61*(2), 289-299.

Song, J.H.; Furman, K.C.. A maritime inventory routing problem: Practical approach. *Comput. Oper. Res.* **2013**, *40*(3), 657-665.

Subramanian, K.; Maravelias, C.T.; Rawlings, J.B. A state-space model for chemical production scheduling. *Comput. Chem. Eng.* **2012**, *47*, 97-110.

Sundaramoorthy, A.; Maravelias, C.T. Computational Study of Network-Based Mixed-Integer Programming Approaches for Chemical Production Scheduling. *Ind. Eng. Chem. Res.* **2011a**, *50*(9): 5023-5040.

Sundaramoorthy, A.; Maravelias, C.T. A General Framework for Process Scheduling. *AIChE J.* **2011b**, *57*(3): 695-710.

Velez, S.; Maravelias C.T. Mixed-Integer Programming Model and Tightening Methods for Scheduling in General Chemical Production Environments. *Ind. Eng. Chem. Res.* **2013**, *52*(9): 3407-3423.

Velez, S. Models and Solution Methods for Chemical Production Scheduling. Doctoral thesis, University of Wisconsin-Madison, **2014**.

Velez, S.; Dong, Y.; Maravelias, C.T. Changeover Formulations for Discrete-Time Mixed-integer Programming Scheduling Models. *Eur. J. Oper Res.* **2017**, doi: 10.1016/j.ejor.2017.01.004.

Verderame, P.M.; Floudas, C.A. Operational planning framework for multisite production and distribution networks. *Comput. Chem. Eng.* **2009**, *33* (5), 1036-1050.

Verderame, P.M.; Elia, J.A.; Li, J.; Floudas, C.A. Planning and scheduling under uncertainty: a review across multiple sectors. *Ind. Eng. Chem. Res.* **2010**, *49(9)*, 3993-4017.

Vidal, C.J.; Goetschalckx, M. Strategic production-distribution models: A critical review with emphasis on global supply chain models. *Eur. J. Oper. Res.* **1997**, *98* (1), 1-18.

Vieira, G.; Herrmann, J.W.; Lin, E. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *J. Sched.* **2003**, *6*, 39-62.

Vin, J.P.; Ierapetritou, M.G. Robust short-term scheduling of multiproduct batch plants under demand uncertainty. *Ind. Eng. Chem. Res.* **2001**, *40*(21), 4543-4554.

Viswanathan, S.; Mathur, K. Integrating routing and inventory decisions in one-warehouse multiretailer multiproduct distribution systems. *Manage. Sci.* **1997**, *43* (3), 294-312.

Wolsey, L.A. MIP modelling of changeovers in production planning and scheduling problems. *Eur. J. Oper. Res.* **1997**, *99*(1): 154-165.

Wolsey, L. A. *Integer Programming.* **1998**, Wiley: New York, NY.

Wonnacott, T.H.; Wonnacott, R.J. *Introductory Statistics for Business and Economics.* **1990**, John Wiley & Sons: Toronto, Canada.

You, F.; Grossmann, I.E. Design of responsive supply chains under demand uncertainty. *Comput. Chem. Eng.* **2008**, *32*(12), 3090-3111.

You, F; Pinto, J.M.; Capon, E.; Grossmann, I.E.; Arora, N.; Megan L. Optimal distribution-inventory planning of industrial gases. I. Fast computational strategies for large-scale problems. *Ind. Eng. Chem. Res.* **2011a**, *50* (5), 2910-2927.

You, F.; Pinto, J.M.; Grossmann, I.E.; Megan L. Optimal distribution-inventory planning of industrial gases. II. MINLP models and algorithms for stochastic cases. *Ind. Eng. Chem. Res.* **2011b**, *50* (5), 2928-2945.

Yu, Y.; Chu, F.; Chen, H. A model and algorithm for large scale stochastic inventory routing problem. Proceedings of Service Systems and Service Management International Conference, **2006**, 355–360.

Zhang, Q.; Sundaramoorthy, A.; Grossmann, I.E.; Pinto, J.M. Multiscale production routing in multicommodity supply chains with complex production facilities. *Comput. Oper. Res.* **2017**, *79*, 207-222.

Zentner, M.G.; Pekny, J.F.; Reklaitis, G.V.; Gupta, J.N.D. Practical considerations in using model-based optimization for the scheduling and planning of batch/semicontinuous processes. J. Proc. Control **1994**, *4*(4): 259-280.