INVENTORY ROUTING FOR PERISHABLE PRODUCTS

Thèse présentée en vue de l'obtention du grade de
docteur en sciences économiques et de gestion

Mahmood REZAEI

Promoter:

Yves CRAMA, University of Liège, Belgium

Members of jury:

Yasemin ARDA, University of Liège, Belgium
Dorothée HONHON, University of Texas at Dallas, U.S.A.
Martin SAVELSBERGH, Georgia Institute of Technology Atlanta, U.S.A.
Michael SCHYNS, University of Liège, Belgium
Tom VAN WOENSEL, Eindhoven University of Technology, the Netherlands

To the angels of my life,

My parents, my sons, my wife.

Contents

Acknowledgments

My six-year doctoral study was a challenging period of my academic and yet family life. During all ups and downs of these years, I managed to survive but also thrive. Accomplishing so, however, would not have been possible without others' help. My parents, wife, promoter, and friends have been my companions in this journey.

My very special thanks goes to my promoter, Yves Crama. From the very first day of my arrival in QuantOM research group to the moment that I am writing these sentences, he has been an amazing adviser and friend. And, I hope that he remains an everlasting friend for me. There were plenty of days I had long scientific discussions with him. Knowledgeable, perceptive, and patient are the most relevant terms by which I can describe him. By working with Yves, not only did I add to my knowledge, but also I could strengthen my analytical, teaching, presentation, and writing skills. He has been assertive to criticize my works, but he always did it in an extremely respectful way. Above all I learned from Yves, here is his most precious advice: "Aim high!"

I sincerely thank my thesis committee and jury. I cordially thank Yasemin Arda for her valuable comments on my works, and for understanding and supporting some difficult moments of my life. I thank Dorothée Honhon who patiently scrutinized my works, and smartly revealed their strengths and weaknesses. I highly appreciate the contributions of Tom Van Woensel to my thesis. Tom was a great co-author, who thoughtfully disordered and reorganized my writings, and who willingly hosted my three-month research visit to Eindhoven University of Technology, the Netherlands. I thank Michael Schyns, a genial colleague who always had time for me, who patiently answered my technical questions on coding, and who was generous in telling me the tricks and secrets to improve the speed of my code. I sincerely thank Martin Savelsbergh for his confidence in me and for hosting me at the University of Newcastle, Australia, for two months. Working with Martin was challenging, as he is scientifically strict. However, for a duration of over three years, I eagerly worked with Martin because he was so modest, intimate, and courteous. My single paper with him took three years to come to the end, but turned out to be one of my best scientific works ever, if not the best.

My colleagues and friends in QuantOM research group had an undeniable role to making a pleasant working atmosphere in HEC-ULg. Mutual respect is the most appropriate word to explain the relationship between me and the best friends I made in Belgium: Alessandro, Alireza, Alper, Ann-Sophie, Ashwin, Cédric, Célia, Christian, Christine, Elisabeth, Guillaumme, Hande, Martine, Maud, Sabine, Sara, Stefano, Stéphanie, Véronique, and Virginie. I was so glad to sit in the same office with Thierry Pironet, who was magnanimous in devoting time to help me whenever I needed, and whose laugh always made me laughing.

I highly appreciate the decision makers in the funding sources of FNRS (Fonds National de la Recherche Scientifique), ARD (Administration de la Recherche et du Développement), and PRISME of HEC-ULg for supporting my attendance to the conferences in Belgium, Italy, Portugal, Norway, and USA. They also supported my research visits to the University of Newcastle in Australia, and Eindhoven University of Technology in the Netherlands. My research was also partially supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office (grant P7/36).

I cannot find any appropriate word to thank my parents, who have been my role models in determination, patience, sacrifice, and benevolence. I praise them for their unconditional support to me in every single moment of my life. I also cordially thank my parents-in-law for their enormous trust on me and for their unique role to keep me motivated.

Last but not the least, I thank my wife, Fahimeh, who has never spared any attempt in building a better future for our family and especially our children. I gratefully acknowledge her heartening words and constructive critics. I admire her indefatigable endeavors to improve the quality of our life, and her benevolence view to others' lives.

Thesis Abstract

We explore three problems in this thesis and develop solution methods for each problem. First, an inventory routing problem for a perishable product with stochastic demands is considered and different approximate solution methods are developed to solve. Based on computational experiments, the solution methods are compared in terms of average profit, service level, and actual freshness. The impact of relevant parameters on the performance of the solution methods is investigated. Managerial insights are drawn by analyzing the impact of shelf life and store capacity on the profit. The value of considering uncertainty and the value of accessing full information are measured. The computational results highlight that a simple ordering policy can often replace a more sophisticated solution method, while preserving the same efficacy.

Second, we introduce a vehicle routing problem (VRP) where a set of stores places deterministic orders to a logistics service provider (LSP) for two successive periods. Deliveries requested in each period can be shifted by the LSP to the other period, possibly with modified quantities. The LSP incurs a penalty for any diversion from the initial delivery period. The data regarding shifted delivery quantities and penalties are provided by the stores. From the perspective of the LSP, diversions could be beneficial if savings in the routing costs outweigh the penalties. In this work, we introduce a new two-period VRP model where the LSP seeks to improve its total cost, compared to solving two independent VRPs with fixed delivery periods, by allowing deliveries to be shifted. We solve this model to optimality by an efficient branch-and-price algorithm implementing several cutting-edge techniques. We draw algorithmic and managerial insights based on our test instances.

Third, a two-period VRP is considered where the orders placed by stores for each period can be partially shifted to the other period, given that the sum of the delivery quantities in two periods to each customer is fixed. A linear penalization of delivery shifts is assumed based on the quantity shifted. We represent two mixed integer linear programming (MILP) formulations for the problem. A column-row generation algorithm to solve the LP-relaxation of the first formulation is developed. To solve the LP-relaxation of the second formulation, we develop a column generation algorithm. Details of two label-setting algorithms to solve the pricing problems of the column-row generation and column generation algorithms are discussed. Numerical results can be compared with a similar model in which only full delivery shifts are allowed.

Chapter 1

Introduction

The subject of this thesis falls in the domain of supply chain management (SCM). In SCM, the flow of products and services from origin to point of destination is planned and monitored, with the objective of minimizing costs or maximizing profit. Production systems exploit benefits of SCM by synchronizing the activities of their procurement, production, and sale units. SCM is also applied to service systems, where the coordination of activities in procurement and sale units (both including distribution) is planned. In practice, there are often several points of origin to pick up raw materials, several production units, possibly geographically scattered or dealing with different types of products, and multiple destination points where the final products should be delivered. The assumption of multiplicity of origin and destination points are also valid in many service systems.

An omnipresent application of SCM in service systems is observed in food retail chains, where products are shipped from suppliers to warehouses and from warehouses to stores to make them available to end customers. Consider a simplified version of the addressed SCM service system with a single origin point (supplier or warehouse) and a set of stores as destination points. Assume that the only decisions to be made in such a simplified system are on inventory (how much and how often to deliver to each store) and distribution (how to dispatch vehicles to deliver to stores). Devising a centralized decision making model in order to optimize total costs or total profits of this network is not a trivial task, and often needs some additional simplifying assumptions. In food retail chain systems, there are often more features which add to the complexity of the problem, such as stochastic demands from end customers and multiple dry and fresh products. Despite imposing simplifying assumptions, mathematical models are valuable because of providing insights for practitioners and managers. The more realistic the assumptions are, the more reliable the insights are.

1.1 Research incentives

The research carried out in this thesis is motivated by the inventory control of perishable products in food retail industry, where multiple perishable products each with a limited shelf life are distributed from a central warehouse to a set of geographically dispersed sale points. This has been a crucial decision-making problem for retail chain managers worldwide for decades, due to a very narrow profit margin in food retail industry. Although the problem has been studied by many researchers, but retail managers always look for new opportunities to increase revenue and service level, decrease costs, and consequently improve the profit.

Two of the most common inventory control policies in food retail chains are simplified versions of retail managed inventory (RMI) policy and periodic deliver-up-to-level (PUL) policy. Both policies need to decide on delivery quantities and delivery routes, where each delivery route is a sequence of stores to be visited by a vehicle. In the RMI policy, each store applies its own inventory control system and places its orders independently from other stores with the objective of minimizing long-term inventory costs. In this system, either the transportation costs are completely neglected or a fixed delivery cost is considered as the transportation cost incurred by each store when it is visited. Then, from the perspective of a logistics service provider (LSP), decisions are made on delivery routes, i.e. transportation decisions, in each period with the objective of minimizing transportation costs in the current period. Under a PUL policy, however, a centralized decision making system decides how often each store should be visited depending of how small or big its daily demand is. Big stores are visited every day, whereas small stores are visited every other day or every three days. The delivery quantity in each visit is up to a predetermined level, e.g. store capacity. Then, a transportation fleet is used to plan delivery routes in each period such that the transportation cost in the current period is minimized.

The mentioned policies are simple to implement and abundant in practice but not efficient enough in terms of overall long-term inventory and routing costs. The inefficiency is mainly due to the lack of synchronization between inventory and transportation-related decisions. Many other relevant works in the literature which take into account the synchronization, neglect perishability of products or stochasticity of demand from end customers. In this sense, there is a research gap between the real problem and the problems addressed in the literature.

1.2 Research questions and objectives

The main research questions of this thesis are:

- Can we formulate an optimization model to help retail chain managers to make decisions on inventory control and routing of perishables?
- Can the model be solved by the existing solution methods in the literature? If not, what are the shortcomings? Can we develop new solution methods to solve it? How do they compare?
- What insights can be drawn out of the problem and by applying the solution method(s)?

The main objective of this research is to provide useful insights for supply chain managers in food retail chains on inventory control and distribution of fresh products. More specifically, we aim at providing a practical way about how such a decision making problem can be formulated as a mathematical model, and how the resultant model can be solved. The mathematical models we develop in this thesis may not perfectly match the real problem that a food retail chain faces. This is because inventory control and distribution policies are different in retail chains. However, similar to other scientific works, this research provides some generic models which can be viewed as benchmarks for many retail chains whatever their inventory control and distribution policies are.

From a mathematical point of view, a very essential objective of this research is to synchronize the inventory and routing-related decisions. A centralized inventory control system to manage inventory levels at a set of stores is know under the appellation of vendor-managed inventory (VMI) in the literature. Perhaps the best known mathematical problem in the framework of a VMI policy is the inventory routing problem (IRP), where a centralized decision making system is established to coordinate delivery quantities and delivery routes to a set of geographically scattered stores during a planning horizon, so that the overall inventory and routing costs are minimized and none of the stores runs out of inventory. In the stochastic IRP (SIRP), where demands from end customers are uncertain, the no-shortage constraint is replaced by satisfying a service level. This thesis studies an SIRP for fresh products, i.e., products with a limited shelf life. We consider realistic assumptions in inventory routing of products in retail chains. These include stochasticity of demands from end customers, perishability of products, and high service level.

1.3 Research restrictions

In a food retail chain, fresh products are usually shipped together in refrigerated trucks. This implies that inventory and routing decisions of several fresh products should be coordinated, i.e., an SIRP for multiple fresh products should be developed. Due to the complexity of the real problem, we confine ourselves to a single perishable product, as is the case in the main body of the literature. Nonetheless, careful analysis of an SIRP for fresh products may provide useful insights which are also applicable to multiple products. Indeed, this research has led to some of these insights.

1.4 Structure of the thesis

In order to address the research questions and achieve the research objectives, we have conducted an extensive literature review in Chapter 2 on the most relevant topics to this research. More specifically, we discuss similarities and differences among the VRP-type problems. In Chapter 3, we define a classical SIRP for perishables with a constraint on the target service level. We develop different solution methods to solve the problem and compare them in terms of average profit, service level, and actual freshness. As part of the most efficient solution method to solve the SIRP, we have to solve a two-period VRP with deterministic demands. Although Chapter 3 deals with solving the latter problem heuristically, a more formal definition and formulation of it is presented in Chapter 4, where an exact solution method is developed to solve it while considering an additional assumption. Indeed, in Chapter 4, a VRP is considered where there are initial deterministic quantities to be delivered to a set of stores in two successive periods. The initial delivery in each period can be fully shifted to the other period, possibly with modified quantities. A penalty is incurred for any diversion from the initial delivery period. The objective is minimizing the total routing and shifting costs. A branch-and-price algorithm is developed to solve the model to optimality. In Chapter 5, we define the same two-period VRP as in Chapter 3 with the possibility of shifting the deliveries partially, rather than fully. We assume that the sum of the delivery quantities in two periods is fixed and shifting penalties are proportional to the quantity shifted. Two mixed integer linear programming (MILP) formulations are presented. A column-row (resp., column) generation algorithm is developed to solve the LP-relaxation of the first (resp., second) MILP formulation. Finally, Chapter 6 concludes the thesis and discusses further research perspectives.

Chapter 2

A classified literature review

2.1 Introduction

In this chapter, we discuss VRP-type problems, namely, the classical VRP, multi-period VRP, periodic VRP, and inventory routing problem. The underlying assumptions for each problem are stated, and the main parameters and decision variables are introduced. We also review the inventory routing problem and inventory control of perishables.

2.2 Vehicle routing problem

The vehicle routing problem (VRP) is defined as the problem of designing delivery routes to a set of geographically scattered stores, subject to side constraints, with the objective of minimizing total routing costs. Each store places an order which must be delivered through a single visit. Each route starts from and ends at a depot and includes a set of customers. By the VRP we mean the capacitated VRP with time windows, where each vehicle has a limited capacity and each store must be visited within an acceptable time window. The time window also applies to the depot, which shows the earliest departure time from and the latest arrival time to the depot. The VRP is defined as follows:

Given:
- $G = (V,A)$: a complete directed graph, where $V = \{0, 1, \cdots, n\}$ represents a depot and a set of geographically scattered stores, and $A = \{(i, j) \in V, i \neq j\}$ is a set of arcs (i, j) linking stores i and j,
- c_{ij}: cost of traversing arc (i, j),
- d_i: demand of store i,
- $[e_i, l_i]$: time window during which customer i must be visited,
- K: number of vehicles each with capacity Q.

Find:
- a set of routes, where each route is a sequence of customers.

Such that:
- each vehicle is able to perform one route,
- each route starts from and ends at the depot and includes each customer at most once (route elementarity),

- each customer is visited by exactly one vehicle (non-split deliveries),
- each customer is visited within its time window,
- the sum of the demands of customers included in each route does not exceed the vehicle capacity Q,
- the number of routes does not exceed the total number of vehicles available K.

Objective:
- total cost of traversed arcs (to be minimized).

A fixed cost may be considered when using each vehicle. If so, the objective function is total cost of traversed arcs and fixed costs. There are two principal mathematical formulations of the VRP: arc-flow formulation and route-based formulation. In an arc-flow formulation, the binary decision variables are defined on arcs, where x_{ij} takes value 1 if arc (i, j) is used in the solution and 0 otherwise. The constraints include: (1) flow-conservation constraints (input flow to each customer must be equal to output flow from it), (2) vehicle capacity constraints (total load on each vehicle cannot exceed its capacity), (3) time window constraints (each customer as well as the depot must be visited within its time window), (4) sub-tour elimination constraints (a route cannot start from and end at any node but the depot), (5) number of vehicles (total output flow from the depot cannot exceed the fleet size). The pitfall of an arc-flow formulation is that it contains a huge number of sub-tour elimination constraints, i.e., one constraint for each subset of customers. However, the arc-flow formulation is used when one aims to apply a branch-and-cut (b&c) algorithm to solve it.

In a route-based formulation, on the other hand, the binary decision variables are defined as feasible routes, where x_r takes value 1 if feasible route r is used in the solution and 0 otherwise. The feasibility of a route is defined in terms of starting from and ending at the depot while visiting each customer at most once (elementarity), respecting vehicle capacity, and respecting time windows. Consequently, the constraints include: (1) set-partitioning constraints (exactly one of the routes covering customer i must be used in the solution), (2) number of vehicles (total number of routes used in a solution cannot exceed the fleet size). The set-partitioning constraints can be replaced by set-covering constraints (at least one of the routes covering customer i must be used in the solution) without affecting the single visit requirement to each customer, while being more efficient in terms of convergence [Feillet, 2010]. This is due to the fact that the dual prices associated with the set-covering constraints are non-negative, whereas those associated with the set-partitioning constraints are free in sign. The pitfall of a route-based formulation is that it lends itself to column generation algorithms which contain a huge number of decision variables, i.e., feasible routes, especially when vehicle capacity is large and time windows are not very restrictive. The advantage, however, is that column generation-based algorithms, branch-and-price algorithm (b&p), or branch-and-price-and-cut algorithm (b&p&c) developed to solve a route-based formulation, usually provide reasonably tight lower bounds as compared to the algorithms developed to solve arc-flow formulations.

In practice, several variants of the VRP have been developed due to the diversity of underlying assumptions and constraints faced in real-life applications. These include the VRP with multiple use of vehicles [Azi et al., 2010], the VRP with split deliveries [Archetti et al., 2006a,b], the VRP with pick-up and deliveries [Dell'Amico et al., 2006], the VRP with selective pick-up and deliveries [Gutierrez-Jarpa et al., 2010], and the multi-depot VRP [Cordeau et al., 1997]. Therefore the VRP may be regarded as a class of problems. A rich body of scientific literature has been developed on the

VRP variants, and successful implementation of practical solution methods have been carried out by researchers from heuristics to Metaheuristics and from Matheuristics to exact methods. The reader is referred to Laporte [2009] as a great survey on exact methods, heuristics, and Metaheuristics to solve VRPs, and Archetti and Speranza [2014] as a specialized survey on Matheuristics to solve the VRP and some of its variants. The most successful exact methods to solve the VRP are column generation-based algorithms which use the route-based formulation, namely b&p and b&p&c algorithms. The most efficient b&p algorithms are developed in Righini and Salani [2006]; Feillet et al. [2004]; Jepsen et al. [2008]; Baldacci et al. [2011a]; Lübbecke and Desrosiers [2005]. As the classical VRP is not the main focus of this research, we avoid further discussion on it.

2.3 Multi-period vehicle routing problem

The multi-period vehicle routing problem (MPVRP) is an extension of the VRP where each customer places an order which must be served within a *period window*, i.e., a set of consecutive periods during a finite planning horizon T. In other words, each order i has a release date (rd_i) and a due date (dd_i), which represents, respectively, the earliest period and the latest period in which the order can be delivered to the associated customer. The MPVRP is defined as follows:

Given:
- T: length of the planning horizon,
- $G = (V, A)$: a complete directed graph, where $V = \{0, 1, \cdots, n\}$ represents a depot and a set of geographically scattered customers, and $A = \{(i, j) \in V, i \neq j\}$ is a set of arcs (i, j) linking customers i and j,
- c_{ij}: cost of traversing arc (i, j),
- d_i: demand of customer i,
- $[rd_i, dd_i]$: period window during which customer i must be visited,
- $[e_i, l_i]$: time window of customer i,
- K: number of vehicles each with capacity Q.

Find:
- a set of routes for each period, where each route is a sequence of customers.

Such that:
- each vehicle is able to perform one route per period,
- each route starts from and ends at the depot and includes each customer at most once (route elementarity),
- each customer is visited exactly once within its period window and by exactly one vehicle (a single non-split delivery during the period window),
- the sum of the demands of customers included in each route does not exceed the vehicle capacity Q,
- the time window for each store is respected if it is visited,
- the number of routes in each period does not exceed the total number of vehicles available K.

Objective:
- total cost of traversed arcs during the planning horizon (to be minimized).

Similar to the VRP, arc-flow and route-based formulations are used to deal with an MPVRP, and the same decision variables can be exploited by adding a time dimension. In other words, in an arc-flow formulation the binary decision variable x_{ijt} takes value 1 if arc (i, j) is used in period t in the solution and 0 otherwise. Similarly, in a route-based formulation the binary decision variable x_{rt} takes value 1 if feasible route r is used in period t in the solution and 0 otherwise. In an arc-flow formulation, the constraints for each period are: (1) flow-conservation, (2) vehicle capacity, (3) time window (if applicable) or the total travel time for each route, (4) sub-tour elimination, (5) number of vehicles. Moreover, each customer must be visited exactly once during its period window, say, (6) period window constraints. In a route-based formulation, the constraints are: (1) set partitioning (or set covering) constraints, i.e., only one of routes covering customer i is used during the entire planning horizon, (2) number of vehicles, i.e., at most K vehicles are used in each period. All other types of constraints are handled while generating feasible routes for each period. Route r in period t is feasible given that it includes a subset of customers for which t is included in their period windows, respects the time window of each customer, respects the vehicle capacity, and is elementary.

In an MPVRP, the delivery is guaranteed within the period window. In addition to the period window, each customer may impose a time window within which it must be visited, whatever the visiting period is. A Logistics Service Provider (LSP) decides on the delivery period to each customer and the delivery routes in each period, given that the delivery quantities are fixed and non-split. The length of the period window depends on the level of service to be provided to each customer. Contracts are often established between supplier and customers whose cost depends on the length of the period window; the narrower the period window, the higher the service level and the more expensive the contract [Athanasopoulos and Minis, 2013; Archetti et al., 2015a].

Applying efficient exact algorithms to solve the classical MPVRP is quite straightforward. As a case in point, Athanasopoulos and Minis [2013] develop a b&p algorithm for the MPVRP. An elementary shortest path problem with resource constraints (ESPPRC) is solved for each period to identify new routes with negative reduced costs.

Similar to the VRP, extensions of the MPVRP have been examined by researchers. Mancini [2016] considers a MPVRP with additional features such as heterogeneity of fleet and multiple depots. He develops a mixed integer programming (MIP) formulation of the problem, and exploits the adaptive large neighborhood search (ALNS) heuristic to solve it. Archetti et al. [2015a] present different arc-flow formulations, enhanced by valid inequalities, for an MPVRP. Their setting includes not only regular customers with due dates within the planning horizon, but also a set of customers with due dates beyond it. The problem formulation allows postponing service of the latter set of customers to some unknown period beyond the planning horizon at a cost, i.e., they do not have to be served. For such customers, postponement penalties proportionate to the length of their period windows are taken into account. The models are solved using a commercial b&c solver.

Dynamism has different meanings in the VRP context. In one definition of a dynamic MPVRP (DMPVRP), customers place orders dynamically during the planning horizon. For instance, Wen et al. [2010] and Albareda-Sambola et al. [2014] consider a DMPVRP, where at the end of each period, exact information about the orders placed in that period and earlier periods is available, and partial information about the orders upcoming in subsequent periods is gradually revealed. Wen et al. [2010] formulate the problem as an integer non-linear programming (INLP) problem and propose a heuristic method to solve it. They consider the weighted linear combination of three objective functions to be minimized, namely, total travel time, total number of waiting periods, and unbalanced workload in different periods. Albareda-Sambola et al. [2014] develop a formula for the DMPVRP to measure the approximate profit of serving each customer in the current period. In order to decide which customers should be served in the current period, they formulate a VRP where the objective function accounts for profit collection and routing costs. Angelelli et al. [2007a] and Angelelli et al. [2007b] handle a DMPVRP with a single uncapacitated vehicle where, in each period, a set of orders appear. The release date for orders is 1 and the due date is either 1 or 2. Demands with due date 2 can be interpreted as regular demands while demands with due date 1 are urgent demands. The authors consider a planning horizon of two periods and analyze the competitive ratio of three simple heuristics to determine which orders should be delivered in period 1. The heuristics are: (a) deliver orders with due date 1, (b) deliver orders with due date 1 or 2, and (c) deliver orders with due date 1 and a subset of orders with due date 2 which are close to the orders with due date 1. The competitive ratio is a measure of quality of an online algorithm, in terms of a ratio between the value of the solution computed by the algorithm and the value of the optimal solution that can be obtained when demands are known over the whole planning horizon. Angelelli et al. [2009] analyze a similar DMPVRP and develop a variable neighborhood search (VNS) heuristic to solve the problem. Contrary to Angelelli et al. [2007a] and Angelelli et al. [2007b] where the planning horizon is two periods, Angelelli et al. [2009] consider a longer planning horizon and analyze the impact of short term strategies on the average operational costs per period.

Andreatta and Lulli [2008] consider an MPVRP with stochastic demands and a single uncapacitated vehicle. Each node in the network has a known stationary probability to have an urgent request (to be satisfied in the same period), a regular request (to be satisfied in either the same period or the subsequent period), or no request. The objective is to minimize the expected cost per period of serving all demands. The authors use a Markov decision process (MDP) to solve the problem. Each state of the system shows the nodes with urgent requests, the nodes with postponed regular requests, and the nodes with newly placed regular requests. Given the current state of the system, the decision is which newly placed regular requests (along with urgent and postponed requests) should be served in the current period and in which order. Table 2.1 presents some of the MPVRP publications most relevant to this thesis, their characteristics, and the solution methods applied to solve the problems.

2.4 Periodic vehicle routing problem

The periodic VRP (PVRP) is a generalization of the VRP in which vehicle routes must be constructed over a planning horizon of T periods, during which a total demand of W_i units must be delivered to customer i. There is a set of feasible delivery schedules S_i for customer i [Beltrami and Bodin, 1974; Christofides and Beasley, 1984], and one of these schedules must be assigned to i. Assigning a schedule to a customer implies that the customer will receive service in the periods specified in that schedule. For example, $S_1 = \{\{Monday, Wednesday, Friday\}, \{Tuesday, Thursday, Saturday\}\}$ implies that customer 1 can be visited either on Mondays-Wednesdays-Fridays or on Tuesdays-Thursdays-Saturdays in a one-week planning horizon, and the chosen schedule is repeated every week. Let $|s_i|$ indicate the number of visits to customer i according to schedule s_i. Then, assigning delivery schedule s_i to customer i implies that the delivery quantity to customer i at each visit is $\frac{W_i}{|s_i|}$. The PVRP is defined as follows:

Table 2.1: Multi-period vehicle routing models and solution methods: finite planning horizon

Study	Demand	Number of vehicles	Maximum length of period window	Length of the planning horizon	Solution method
Angelelli et al. [2007a]	dynamic	single	2	2	heuristic
Angelelli et al. [2007b]	dynamic	single	2	2	heuristic
Angelelli et al. [2009]	dynamic	K	2	$T > 2$	VNS
Andreatta and Lulli [2008]	dynamic	single	2	infinite	MDP
Wen et al. [2010]	dynamic	K	> 2	$T > 2$	INLP formulation; heuristic
Athanasopoulos and Minis [2013]	deterministic	K	> 2	$T > 2$	b&p
Albareda-Sambola et al. [2014]	dynamic	K	> 2	$T > 2$	ILP formulation; VNS
Archetti et al. [2015a]	deterministic	K	> 2	$T > 2$	commercial b&c solver
Mancini [2016]	deterministic	K	> 2	$T > 2$	MIP formulation; ALNS

Given:
- T: length of the planning horizon,
- $G = (V,A)$: a complete directed graph, where $V = \{0, 1, \cdots, n\}$ represents a depot and a set of geographically scattered customers, and $A = \{(i, j) \in V, i \neq j\}$ is a set of arcs (i, j) linking customers i and j,
- c_{ij}: cost of traversing arc (i, j),
- W_i: total demand of customer i for the entire planning horizon,
- S_i: set of feasible delivery schedules for customer i,
- $[e_i, l_i]$: time window of customer i,
- K: number of vehicles each with capacity Q.

Find:
- an allocation of a feasible schedule s_i in S_i to customer i,
- a set of routes for each period, where each route is a sequence of customers.

Such that:
- each vehicle is able to perform one route per period,
- each route starts from and ends at the depot and includes each customer at most once (route elementarity),
- each customer is visited at most once in each period (at most one non-split delivery in each period),
- number of visits to customer i during the planning horizon is $|s_i|$,
- delivery quantity at each visit to customer i is $\frac{W_i}{|s_i|}$,
- the sum of demands of the customers included in each route does not exceed the vehicle capacity Q,
- the time window for each store is respected if it is visited,

- total number of routes in each period does not exceed the total number of vehicles available K.

Objective:
- total cost of traversed arcs during the planning horizon (to be minimized).

In some PVRPs, an explicit enumeration of delivery schedules for each customer is not given. Instead, it is assumed that each customer i must be visited at a predetermined frequency f_i, i.e., customer i must be visited f_i times during the planning horizon. A fraction $\frac{1}{fi}$ of the total demand has to be delivered to customer i at each visit, i.e., $d_i = \frac{W_i}{f_i}$ is the delivery quantity at each visit [Francis et al., 2008]. Such problems may enforce constraints on the minimum and maximum required spacing between deliveries [Gaudioso and Paletta, 1992]. When the minimum and maximum spacing are the same, say m_i, it implies that deliveries to customer i must occur every m_i periods Cordeau et al. [1997]. In this case, the planning horizon T is the least common multiple of the m_i values. Hence, $\frac{T}{m_i}$ is an integer number for all i and represents the number of times customer i is to be served over the planning horizon. The key characteristic of the PVRP is that the delivery quantity to each customer only depends on the number of visits, i.e., the delivery quantity is $\frac{W_i}{|s_i|}$ or $\frac{W_i}{f_i}$ or $\frac{m_i \cdot W_i}{T}$. In fact, all these representations can be viewed as specific cases of the PVRP definition with feasible delivery schedules introduced earlier.

Customer capacity is not a constraint in the PVRP models, but it is implicitly applied when determining the frequency of visits or when enumerating the feasible delivery schedules. In many PVRP applications, customers with larger demands (or smaller storage capacities) require to be visited more frequently as compared to customers with smaller demands (or larger storage capacities). The examples include grocery distribution, soft drink industry, and waste collection [Hemmelmayr et al., 2009a]. When dealing with a PVRP with given delivery schedules, most of the works assume that the number of visits in each delivery schedule, $|s_i|$, is a fixed number for each customer i. In other words, all s_i in S_i have the same number of delivery visits.

Most of the PVRP papers use an arc-flow formulation, where the decision variables are: x_{ijk}^t taking value 1 if vehicle k traverses arc (i,j) in period t and 0 otherwise, and y_{ik}^s taking value 1 if vehicle k visits customer i on schedule s and 0 otherwise. Note that, in general, using index k for vehicles in VRP-type formulations dismisses the necessity of including sub-route elimination constraints. Some formulations use aggregated versions of the above variables as follows: $\tilde{x}_{ik}^t = \sum_{j \in V} x_{ijk}^t$ taking value 1 if vehicle k visits customer i in period t and 0 otherwise, and $\tilde{y}_i^s = \sum_{k \in K} y_{ik}^s$ taking value 1 if customer i is visited on schedule s. See Francis et al. [2008] for more details on the arc-flow problem formulations. The route-based formulation defines z_r^t taking value 1 if route r in period t is in the solution and 0 otherwise, and $\tilde{y}_i^s$ taking value 1 if customer i is visited on schedule s and 0 otherwise. The latter formulation is seen in Baldacci et al. [2011b].

Variants of the PVRP exist in the literature. The periodic traveling salesman problem (PTSP) is a special case of the PVRP restricted to one vehicle. The PVRP with time windows (PVRPTW) generalizes the PVRP to include time windows for deliveries to the customers. Cordeau et al. [1997] demonstrate that the MDVRP [Cordeau et al., 1997; Vidal et al., 2012] can be viewed as a special case of the PVRP by considering each of T depots to be a day on a T-day planning horizon, and each customer to require one delivery over that horizon. Recall that in the MDVRP, each vehicle is assigned to one depot from a set of depots, and the planning horizon is restricted to a single day.

The PVRP arises in a variety of applications which can be classified into: (a) pickup, (b) delivery, (c) pickup-delivery, and (d) routing for on-site service. Some applications with pickup are: garbage collection where customers have the same frequency with sparsely distributed customers or big street containers, waste collection at hospitals or industrial sites, collection of recyclable materials, and goat milk collection. The applications with delivery include: Coca-Cola products (some customers once a week some twice a week), grocery customers (higher frequency for large customers lower for small customers), hospital linens, vending machine stock. An application with pickup-delivery is inter-library loan services. The applications with routing for on-site service include: maintenance crews, quality inspectors, teaching assistants, and home health care nurses [Campbell and Wilson, 2014].

Variants of the classical PVRP have been defined in the literature. As an instance, Gaudioso and Paletta [1992] formulate a PVRP with the objective of minimizing the maximum number of vehicles employed simultaneously over the planning horizon, i.e., the fleet size, where having multiple routes per vehicle is possible. They also impose additional constraints on the minimum and maximum spacing between two successive deliveries to each customer. They propose heuristic algorithms to solve the problem.

Heuristics have been used in the early works on the PVRP to solve the problem. Beltrami and Bodin [1974] introduces two heuristic ideas: (1) route customers using a Clarke and Wright procedure, then assign routes to days, and (2) assign customers to delivery days randomly, then create routes for each day based on this assignment. In the heuristic presented by Russell and Igo [1979], the authors cluster customers that are close together and that have the same weekly delivery requirements. Then, they consider three heuristics: (1) assign all daily customers to each day of the week, then schedule the remaining customers based on the estimated costs of combining with customers already scheduled on those days, (2) use link exchanges to improve the initial solution, (3) use Clarke and Wright procedure to enforce the spacing of periodic deliveries throughout the week. Christofides and Beasley [1984] offer an exact formulation, but solve the PVRP via a heuristic: assign customers to days, then solve the resulting daily VRPs. The initial assignment in their heuristic is based upon an ordering of customers where those with fewer delivery combinations and larger delivery quantities are scheduled first. Then, customers yielding the smallest increase in total costs are inserted using the allowable delivery combinations.

Metaheuristics have been developed to solve PVRPs. These include: Tabu Search (TS) [Cordeau et al., 1997], Variable Neighborhood Search (VNS) [Hemmelmayr et al., 2009a; Pirkwieser and Raidl, 2010], Genetic Algorithm (GA) [Vidal et al., 2012], and Ant Colony Optimization (ACO) [Matos and Oliveira, 2004].

Few papers exploit exact methods to solve PVRPs. Francis et al. [2006] present an IP formulation in which the frequency of visits, f_i, is a decision variable, generalizing the definition of periodicity in most of the PVRP papers (i.e., a fixed frequency for all delivery schedules of a customer). They create two different relaxations of the problem by separating the decision variables on schedules from those pertinent to routes. Then, they use sub-gradient optimization on a Lagrangian function incorporating both of these relaxations to calculate lower bounds for the objective function. Finally, a b&b algorithm is used to close the optimality gap. Mourgaya and Vanderbeck [2007] develop a model which simultaneously addresses two objective functions: balancing the load on vehicles and creating routes that keep drivers in familiar areas. A Dantzig-Wolfe reformulation and column generation are used to solve the relaxed problem, where the authors use insertion heuristics to price out columns. After solving the LP relaxation at the root node, the resulting solution is rounded to produce a feasible solution to the PVRP by heuristically exploring the b&b tree. Baldacci et al. [2011b] have arguably developed the most successful exact solution method for the PVRP. They present a new IP formulation for the problem and three relaxations based on the formulation which are used to generate strong lower bounds. Along with information from a related dual solution, these lower bounds are exploited to reduce the solution space in such a way that no optimal integer solutions are eliminated. This procedure results in a tractable IP which can be solved to optimality using a commercial solver.

Interested readers are referred to Campbell and Wilson [2014] and Francis et al. [2008] as excellent review papers on the PVRP and its variants. Some of the most important PVRP works we have found in the literature are summarized in Table 2.2.

Table 2.2: Periodic vehicle routing models and solution methods: deterministic demand, single product, finite planning horizon

Study	Periodicity	Other characteristics	Solution method
Pirkwieser and Raidl [2010]	S_i		VNS
Hemmelmayr et al. [2009a]	f_i		VNS
Cordeau et al. [1997]	S_i		TS
Gaudioso and Paletta [1992]	m_i	minimizing number of vehicles	heuristic
Beltrami and Bodin [1974]	S_i		heuristic
Christofides and Beasley [1984]	S_i		IP formulation; heuristic
Vidal et al. [2012]	S_i		GA
Matos and Oliveira [2004]	S_i		ACO
Francis et al. [2006]	S_i	service frequency is a decision variable; f_i is the minimum required service frequency; Lagrangian relaxation to find a LB	b&b
Mourgaya and Vanderbeck [2007]	S_i	balancing the load on vehicles; keeping drivers in familiar areas	CG; b&b
Baldacci et al. [2011b]	S_i		CG; strong LB; commercial solver

2.5 Inventory routing problem

Vendor-managed inventory (VMI) system is an example of successful and still promising business practice. The VMI relies on the cooperation between a supplier and its customers, where information about demand and inventory is shared by the customers with the supplier. Under a VMI system, the supplier takes over the responsibility of managing the inventory of the customers by deciding on replenishment quantities and delivery periods. The consequences are beneficial for both parties: customers need to employ less resources for controlling their inventory, and the supplier has more flexibility for integrating the replenishment quantities and periods to different customers and therefore smooth its production, inventory, and distribution systems [Desaulniers et al., 2016]. The VMI is a managerial policy, which is to be contrasted with retailer-managed inventory (RMI); see Archetti and Speranza [2016]. In an RMI system, the customers decide when and how much to order, independently of each other. In such traditional distribution system, the power of the supplier to optimize the distribution is strongly constrained by the decisions made by its customers, even when the objective is to minimize the transportation cost only [Bertazzi and Speranza, 2012].

Looking at the RMI from an LSP's perspective takes us to the VRP, MPVRP, and PVRP, whereas following a VMI policy leads to an inventory routing problem (IRP) as one of the models under this policy. The classical IRP deals with deterministic demands, whereas the stochastic IRP (SIRP) refers to an IRP with stochastic demands. In the sequel we first deal with the IRP and then review the SIRP.

2.5.1 Deterministic inventory routing problem

The IRP is concerned with the distribution of a single product from a single depot to a set of customers with deterministic demands over a given planning horizon. The objective is to minimize the distribution and inventory costs during the planning period without causing stock-outs at any of the customers. The main decisions in an IRP are: (a) when to serve each customer, (b) how much to deliver to a customer when it is visited, and (c) which routes to use. A more mathematical definition of the IRP is as follows [Coelho et al., 2014a].

Given:
- T: planning horizon,
- $G = (V,A)$: a complete directed graph, where $V = \{0,1,\cdots,n\}$ represents a depot and a set of geographically scattered customers, and $A = \{(i,j) \in V, i \neq j\}$ is a set of arcs (i,j) linking customers i and j,
- c_{ij}: cost of traversing arc (i,j),
- I_{i0}: initial inventory at the depot (I_{00}) and at customer i,
- d_{it}: demand of customer i in period t,
- h_i: unit inventory holding cost at the depot (h_0) and at customer i,
- C_i: store capacity at customer i,
- B_t: quantity of the product made available at the supplier at the beginning of period t,
- $[e_i,l_i]$: time window of customer i,
- K: number of vehicles each with capacity Q.

Find:
- delivery quantity to each customer in each period (note that when delivery quantity to customer i in period t is zero, then customer i is not visited in period t due to the triangular inequalities on distances),
- a set of routes for each period, where each route is a sequence of customers.

Such that:
- each vehicle is able to perform one route per period,
- each route starts from and ends at the depot and includes each customer at most once (route elementarity),
- each customer is visited at most once in each period (at most one non-split delivery

in each period),
- the sum of the delivery quantities to customers included in each route does not exceed vehicle capacity Q,
- the time window for each store is respected if it is visited,
- total number of routes in each period does not exceed the total number of available vehicles K,
- inventory level of each customer at the end of a period does not exceed its inventory capacity,
- inventories at customers are not allowed to be negative, i.e., all demand must be met by previous inventory plus deliveries performed during the time period considered,
- total shipments from the depot to all customers in each period do not exceed its inventory level plus the quantity made available in that period.

Objective:
- total inventory and distribution costs during the planning horizon (to be minimized).

Most IRP models are MIP formulations using arc-flow decision variables such as: x_{ijt} takes value 1 if arc (i,j) is used in period t in the solution and 0 otherwise, y_{it} is a non-negative decision variable representing the delivery quantity to customer i in period t. If one aimed to use a route-based formulation in a column generation framework, it would not be easy to determine the feasibility of a route in the pricing problem in terms of respecting vehicle capacity. This is due to the fact that delivery quantities are not know in advance but they are decision variables. The only exact method we have found which uses a route-based formulation is the paper by Desaulniers et al. [2016].

The IRP has a wide range of road-based applications including the distribution of gas [Campbell and Savelsbergh, 2004a; Gronhaug et al., 2010], fuel [Popović et al., 2012], automobile components [Alegre et al., 2007; Stacey et al., 2007], perishable products [Federgruen and Zipkin, 1984; Federgruen et al., 1986], groceries products [Gaur and Ficher, 2004], cement [Christiansen et al., 2011], and blood products [Hemmelmayr et al., 2009b].

Since the IRP has the flexibility to decide how much to deliver to each customer and which routes to use in each period, the decision space becomes enormously large as compared to the VRP, the MPVRP, and the PVRP. Therefore, determining the optimal solution in an IRP is extremely difficult. In most cases, it is quite challenging to find the optimal solution for even very small instances of the IRP [Campbell et al., 1998]. As the IRP is considerably more complex than other VRP variants, researchers have striven to develop solution methods for simplified IRP models rather than generalizing the classical IRP definition. Here, we briefly mention the most common simplifying assumptions on the IRP.

The deliver-up-to-level policy: The deliver-up-to-level (UL) policy is the most-widely used replenishment policy. It simplifies the decisions on the quantities to be delivered when a customer is visited. In the UL policy, each delivery must fill the inventory to its maximum capacity C_i. Hence, the quantity to be delivered to customer i is the difference between its maximum capacity and its current inventory level. As a result, the decisions in an IRP with the UL policy are restricted to: (a) the periods when each customer should be visited, (b) the delivery routes. Contrary to the UL policy, in the maximum-level (ML) policy, any quantity can be delivered as long as the maximum level determined by the customer is not exceeded. The maximum level can simply be the maximum capacity C_i or any other level below C_i. The ML policy clearly encompasses the UL policy and is more flexible, but also more difficult to solve given the extra set of decision variables. The UL policy has been used as a way to simplify the search for good solutions and has been widely studied [Bertazzi et al., 2002; Archetti et al., 2007; Solyali and Süral, 2011]. Other works which do not explicitly consider a UL policy indeed follow an ML policy. In the work of Bertazzi et al. [2002], shipments from the supplier to the customers are performed by a single capacitated vehicle. Each customer determines a minimum and a maximum level of the inventory, and needs to be visited before its inventory hits the minimum level. Every time a customer is visited, the replenishment quantity is such that the maximum level of the inventory is reached at the customer (this is a UL policy). The authors propose a two-step heuristic algorithm to solve the problem. Archetti et al. [2007] consider the same problem defined by Bertazzi et al. [2002] under both UL and ML policies. They present an MILP model and derive new additional valid inequalities used to strengthen the linear relaxation of the model. Archetti et al. [2007] are the first to implement an exact branch-and-cut algorithm to solve the model optimally. Later, Solyali and Süral [2011] develop a stronger formulation under a UL policy and solve it by branch-and-cut.

Direct delivery: Large delivery quantities to customers can encourage the supplier to resort to a direct delivery policy. Gallego and Simchi-Levi [1990] show that in the case of continuous shipping times, the worst-case performance ratio of direct shipping with respect to a lower bound on the optimal cost is less than 1.06, whenever the economic lot size of each customer is at least %71 of the vehicle capacity. When the direct delivery policy is applied to an IRP, the decisions are restricted to: (a) the periods when each customer should be visited, (b) the delivery quantity to each customer. The resultant problem has no routing aspect anymore. However, it is still called the IRP with direct deliveries and definitely falls in the class of the VMI systems, as the limited number of vehicles at hand or the limited supply at the depot needs a sort of coordination between the decisions on delivery periods and delivery quantities to different customers. Bertazzi [2008] analyze different direct shipping policies in an IRP with an unlimited fleet but a limited supply. Unlike Gallego and Simchi-Levi [1990], the deliveries can be performed at discrete time instants in Bertazzi [2008]. He demonstrates that, in the worst-case, the ratio between the cost of the optimal direct shipping policy and the optimal cost of the problem in which routing is allowed is not greater than 2 whenever the unit volume of the product to be delivered to each customer is not lower than 1/4 of the vehicle capacity, and the ratio is 1.2 if the unit volume is not lower than the capacity; more than one visit to each customer is required in the latter case. See Bertazzi and Speranza [2012] for a tutorial on the IRP papers with direct delivery.

Zero inventory delivery: Following this policy, in each period, only the set of customers whose inventory levels are down to zero are replenished. Given a current inventory level, the next delivery period to each customer is easily determined. However, depending on the next delivery quantity, the subsequent delivery periods are not known. Chan et al. [1998] and Jaillet et al. [2002] apply this policy to the IRP. Although this policy maximizes the volume delivered to a customer, it is not necessarily the best option in terms of long-term distribution costs. This is because it does not recognize the synergies that may exist between customers. Indeed, it may happen that two customers that are geographically close (and thus can be served on a single trip) will never be routed together because of their initial inventories and their usage rates (optimal replenishment days differ).

Single vehicle: Dealing with a single vehicle means that only one route per period is created. In this case, the main decisions are: (a) the set of customers to be visited in each period, (b) the delivery quantity to each customer. Once these decisions are made for every period, the routing part is simply a set of independent TSPs to be solved. However, solving the TSPs cannot sequentially follow the first two decisions, as the total cost of the TSPs depends on the customers to be visited and the delivery quantities. Bertazzi and Speranza [2013] formulate the problem as an MILP under an ML policy with arc-flow decision variables and present valid inequalities, but do not develop any solution method. Heuristic algorithms developed to solve the single vehicle IRP include the fast local search of Bertazzi et al. [2002], the hybrid of mathematical programming and tabu search of Archetti et al. [2012], and the adaptive large neighborhood search (ALNS) of Coelho et al. [2012a]. Works using an exact branch-and-cut algorithm to solve an IRP with a single vehicle include Archetti et al. [2007] (under UL and ML policies) and Solyali and Süral [2011] (under UL policy).

Constant demand rate: The consumption rates of the customers are assumed stationary over time under this policy. Some papers assume that the demand of each period must be satisfied at once, so the demand of each period must be available at the beginning of the period [Raa and Aghezaaf, 2008, 2009]. Other works assume that customers deplete inventory on a continuous time basis [Ekici et al., 2015]. Under the latter assumption, not only do the delivery periods matter but also the delivery time is important so that no stock-out occurs during the delivery period.

Periodic delivery: Solving IRPs solely based on cost considerations may lead to inconveniences to both parties in terms of frequency of the deliveries [Coelho et al., 2012b]. Periodic or cyclic routing strategies are schedules for a short-term planning horizon which are assumed to be repeated over the entire planning horizon. The periodicity of the policy implies that the inventory levels at the end of the period must be equal to the initial levels [Bertazzi and Speranza, 2012]. Periodic deliveries are desirable to many customers as they can better organize their human resources when they receive deliveries on a regular basis. Periodic deliveries also lead to less fluctuations in the fleet size, vehicle load, and delivery quantities. When applying a periodic delivery policy to an IRP with constant consumption rates for customers, the resultant problem can be formulated as a PVRP. The constant consumption rates can be used to determine the time between two consecutive visits, m_i, for each customer [Campbell and Wilson, 2014]. The time between two visits should be as long as possible to minimize delivery costs, but should be sufficiently small to keep the customer from running out of product. Once m_i is known, the remaining problem can be solved as a PVRP for a defined planning period. This can be regarded as an OU replenishment policy, where the deliver up-to-level point is not the customer capacity but is calculated based on the delivery frequency. In order to be able to exploit such transformation, the usage rate must be constant and the time between two consecutive visits must be fixed so that the delivery quantity to each customer is a fixed value. Rusdiansyah and Tsao [2005] transform an IRP inspired by the replenishment of vending machines into an instance of the PVRPTW, and Hemmelmayr et al. [2009b] transform an IRP for the blood bank into a PVRP. If demand consumption is not stationary, considering periodicity of delivery in the IRP leads to the periodic IRP rather than the PVRP; see Gaur and Ficher [2004]. The crucial constraint, to be respected in both PVRP and PIRP, is that every customer must have the same inventory level at the beginning of each cycle.

Decomposition approaches: Some of the algorithms proposed in the literature decompose the IRP into two stages: (a) inventory control (determining the delivery amounts), and (b) vehicle routing [Campbell and Savelsbergh, 2004b; Federgruen and Zipkin, 1984; Qu et al., 1999]. In some papers, the overall solution is found by iterating between these two problems [Federgruen and Zipkin, 1984; Qu et al., 1999]. For example, Campbell and Savelsbergh [2004b] develop a two-phase decomposition approach. In phase 1, a few periods of the planning horizon are considered, say the first k days, and periods $k+1$ and onward are aggregated into greater time units, say weeks. Moreover, a small set of good delivery routes are considered and a route-based MIP formulation is solved. The solution to the MIP in phase 1 specifies the delivery quantity to each customer in each of the first k periods. In phase 2, given the delivery quantities, a VRP is solved for each period. The set of good routes in phase 1 is determined by partitioning customers into disjoint good clusters and then developing good routes for each cluster. A good cluster is defined as a group of customers that can be served at low cost by a single vehicle for a long period of time. To this end, the cost of a cluster does not only depend on the geographic locations of the customers in the cluster, but also on whether the customers in the cluster have compatible inventory capacities and usage rates.

The IRP without simplifying assumptions: Algorithms capable of solving the IRP without simplifying assumptions are more recent. These include the ALNS heuristic of Coelho et al. [2012b], the exact branch-and-cut algorithms of Coelho and Laporte [2013a], Coelho and Laporte [2013b], and Coelho and Laporte [2014d], and the branch-and-price-and-cut algorithm of Desaulniers et al. [2016]. All of these papers provide an algorithm to solve the problem under the same assumptions, and are tested on the same set of benchmark instances, thus allowing for a clear comparison of the performance of each algorithm. In the most recent work, Desaulniers et al. [2016] apply a branch-and-price-and-cut algorithm to solve an IRP with an ML inventory replenishment policy. Route delivery patterns are generated, where a route delivery pattern specifies the quantity delivered to each customer along the corresponding route; only extreme route delivery patterns are considered and their convex combinations are used to generate any other route delivery patterns. In the branch-and-bound algorithm, the lower bounds in each node are computed using a column generation algorithm, where one subproblem for each period is solved as an ESPPRC. Cutting planes are added dynamically to tighten the linear relaxations. Violated inequalities are found by enumeration and added to the master problem, and the subproblem is adjusted to include dual variables of the added inequalities and is solved again. The valid inequalities they use include inequalities on the minimum number of visits per customer introduced by Archetti et al. [2007] for the single vehicle case and Coelho and Laporte [2014d] for the multi-vehicle case with an arc flow model, minimum number of routes per time interval, minimum number of sub-deliveries per demand introduced by Desaulniers [2010], and capacity inequalities introduced by Laporte et al. [1985].

The IRP with extra features: Coelho et al. [2012a] introduce the concept of consistency in IRP solutions, in terms of the fleet size, the vehicle load, the frequency of the deliveries, and the quantities delivered. Indeed, having consistency means sacrificing a part of cost improvement with the aim of having a more stable required fleet size, attaining less fluctuations in the vehicle load, and visiting each customer more regularly. The authors formulate the multi-vehicle IRP, with and without consistency requirements, as MILP problems and propose a matheuristic for their solution. Some of the consistency features depend on the stability of the demand. If the demand is highly variable, customers would expect delivery quantities to be variable as well, so consistency in delivery quantities to each customer makes little sense. However, the application of the VMI requires some demand stability, which legitimates the consistency features they propose. Coelho et al. [2012b] introduce the IRP with Transshipment (IRPT). They consider the possibility of performing transshipment between customers so as to further reduce the overall cost. They present a formulation that allows transshipment, either from the supplier to customers or between customers. They also propose an adaptive large neighborhood search heuristic to solve the problem. Their approach solves four different variants of the problem, i.e., the IRP with and without transshipment under UL and ML policies: the IRP-UL, the IRP-ML, the IRPT-UL, and the IRPT-ML.

The interested reader is referred to Andersson et al. [2010] and Coelho et al. [2014a] as excellent review papers on IRPs from the application and the methodological point of views, respectively. For tutorials on the IRP see Bertazzi and Speranza [2011, 2012, 2013]. Table 2.3 shows some of the most important IRP papers we have found under the following assumptions: finite planning horizon, storage capacity, vehicle capacity, single product, discrete and instant consumption of demand.

2.5.2 Stochastic inventory routing problem

The SIRP assumes uncertain future demands for customers, and this is the way it differs from the IRP. In the SIRP, we are given the probability distribution of the demands. While the majority of papers on the SIRP assume that demands in each period are fully realized at the end the period, there are models assuming that demands are realized upon arrival of the vehicle at each customer [Berman and Larson, 2001; Huang and Lin, 2010].

Demand stochasticity means that shortages may occur, and there is often a positive probability that a customer runs out of stock. To discourage the shortages, a penalty is imposed whenever a customer runs out of stock, and this penalty is usually modeled as a proportion of the unsatisfied demand. Unsatisfied demand is typically considered to be lost sale [Minkoff, 1993; Kleywegt et al., 2004], and is rarely dealt with as backlogging, as in the work by Yu et al. [2012]. In either case, penalties may apply. A pre-defined service level may apply too, which imposes a minimum inventory level at each customer in each period [Yu et al., 2012]. The objective is to choose a delivery policy that minimizes the expected cost per unit time when the planning horizon is infinite. So, in both cases, the objective is to minimize the expected cost per unit time. The SIRP is defined as follows:

Given:
- T: planning horizon, possibly infinite,
- $G = (V,A)$: a complete directed graph, where $V = \{0,1,\cdots,n\}$ represents a depot and a set of geographically scattered customers, and $A = \{(i,j) \in V, i \neq j\}$ is a set of arcs (i,j) linking customers i and j,
- c_{ij}: cost of traversing arc (i,j),
- I_{i0}: initial inventory at the supplier (I_{00}) and at customer i,
- D_{it}: random variable representing demand at customer i in period t,
- h_i: unit inventory holding cost at the depot (h_0) and at customer i,
- p_i: unit back-order or lost sale cost at customer i,
- C_i: inventory holding capacity at customer i,
- B_t: quantity of the product made available at the supplier at the beginning of period t,
- $[e_i,l_i]$: time window of customer i,
- K: number of vehicles each with capacity Q.

Table 2.3: Inventory routing models and solution methods: deterministic demand, finite planning horizon, single product, no shortage

Study	Number of vehicles	Repleni. policy	Direct delivery	Periodic delivery	Production supply	Other characteristics	Solution method
Campbell and Savelsbergh [2004b]	K	ML	no	no	unlimited	constant demand rate; multiple tours; short-term planning	route-based MIP formulation; limited number of routes; 2-phase decomposition algorithm
Bard and Nananukul [2010]	K	ML	no	no	limited: decision variable	production, inventory, and routing	b&p
Coelho and Laporte [2014d]	K	ML	no	no	limited		valid inequalities; b&c
Coelho and Laporte [2013a]	K	ML	no	no	limited	consistency features; multi-product	b&c
Coelho et al. [2012b]	single	UL; ML	no	no	limited	transshipments; direct deliveries	ALNS
Coelho et al. [2012a]	K	UL	no	no	limited	consistency constraints	MILP formulation; matheuristic; ALNS
Solyali and Süral [2011]	single	UL	no	no	limited		b&c
Archetti et al. [2007]	single	UL	no	no	limited		b&c
Bertazzi et al. [2002]	single	UL	no	no	limited	min and max level of inventory by each customer	heuristic
Desaulniers et al. [2016]	K	ML	no	no	unlimited	supplier capacity; unit holding cost at supplier and at customers	b&p&c
Gaur and Ficher [2004]	K	ML	no	yes	unlimited	exact method for at most two stores per route; heuristic for more than two stores per route	fixed partition policy; weighted matching; heuristic
Ekici et al. [2015]	K	ML	no	yes	unlimited	constant continuous consumption rate	iterative clustering; heuristic
Raa and Aghezaaf [2009]	K: decision variable	ML	no	yes	unlimited	constant consumption rate; multiple tours per vehicle clustering	heuristic; column generation
Raa and Aghezaaf [2008]	single	ML	no	yes	unlimited	constant consumption rate; multiple tours per period	column generation for distribution patterns
Bertazzi [2008]	K	ML	yes	no	limited	multiple-products; infinite horizon	worst-case analysis

Find:
- delivery quantity to each customer in each period (note that based on the triangular inequalities on distances, when delivery quantity to customer i in period t is zero, the customer is not visited in period t),
- a set of routes for each period, where each route is a sequence of customers.

Such that:
- each vehicle is able to perform one route per period,
- each route starts from and ends at the depot and includes each customer at most once (route elementarity),
- each customer is visited at most once in each period (at most one non-split delivery in each period),
- the sum of the delivery quantities to customers included in each route does not exceed vehicle capacity Q,
- the time window for each store is respected if it is visited,
- total number of routes in each period does not exceed the total number of available vehicles K,
- inventory level of each customer at the end of a period does not exceed its inventory capacity,
- pre-defined service levels (if any) at customers are respected,
- total shipments from the depot to all customers in each period does not exceed its inventory level plus the quantity made available in that period.

Objective:
- expected inventory and distribution costs per period or expected total cost over the planning horizon (to be minimized).

Due to complexity of the SIRP, simplifying assumptions are frequent in the models, as they are in the IRP. These assumptions include considering a single capacitated vehicle [Coelho and Laporte, 2014c; Schwartz et al., 2006; Reinman et al., 1999], a single uncapacitated vehicle [Qu et al., 1999], and direct deliveries [Kleywegt et al., 2002; Reinman et al., 1999; Barnes-Schuster and Bassok, 1997].

Markovian decision process (MDP) is the most widely exploited solution approach to solve an SIRP. Most of the papers that use an MDP to solve an SIRP formulate the cost (or reward) function explicitly for an infinite planning horizon. The underlying assumption for such a formulation is that the demand probability function must be stationary, i.e., the probability distribution of demand at each customer, $Pr(D_i)$, does not depend on the period and has the same parameters in all periods. Under this condition, an optimal itinerary can be calculated which depends only on the current state (and not on the period). The state of the system in each period is defined by the list of inventory levels at customers, i.e., by $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$, where x_i is the inventory level at customer i. In each period, a comprehensive decision regarding which customers to visit, how much to deliver to them, and how to combine customers into delivery routes is called an itinerary [Kleywegt et al., 2004], and is represented by $\boldsymbol{y}$. Let us indicate the vector of delivery quantities to customers (which is part of an itinerary) by $(y_1, y_2, \cdots, y_n)$. The action space for state $\boldsymbol{x}$ is defined as the set of all feasible itineraries for state $\boldsymbol{x}$, i.e., itineraries that satisfy customers' capacities and route constraints in terms of length, time window, vehicle capacity, and number of available vehicles. By $\mathscr{A}(\boldsymbol{x})$, we represent the action space (all feasible itineraries) for state $\boldsymbol{x}$, and by $\mathscr{X} = [0, C_1] \times [0, C_2] \times \cdots [0, C_n]$ we represent the state space, i.e., all feasible itineraries for all states. The cost function for an MDP is defined as follows:

$$V^*(\boldsymbol{x}) = \min_{\boldsymbol{y} \in \mathscr{A}(\boldsymbol{x})} \{c(\boldsymbol{x},\boldsymbol{y}) + R(\boldsymbol{y}) + \sum_{\boldsymbol{z} \in \mathscr{X}} V^*(\boldsymbol{z}) \cdot Pr(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})\} \tag{2.1}$$

where,

$$c(\boldsymbol{x},\boldsymbol{y}) = \sum_{i \in N} \sum_{d_i=0}^{\infty} Pr(D_i = d_i) \cdot [p_i(d_i - x_i - y_i)^+ + h_i(x_i + y_i - d_i)^+] \tag{2.2}$$

and $R(\boldsymbol{y})$ is the routing cost associated with itinerary $\boldsymbol{y}$.

Usually an iterative procedure has to be used to solve problem 2.1. There are standard techniques such as policy iteration, value iteration, and successive approximation. These algorithms are practical only if the state space is small, and the optimization problem on the right-hand side can be solved efficiently. Neither of these requirements are satisfied by practical instances of the SIRP, as the state space is usually extremely large, and problem 2.1 has a vehicle routing problem as a special case, which is NP-hard [Campbell et al., 1998]. Due to such complexity, many researchers use approximations of different functions embedded in the cost function 2.1 or decompose it. The book by Powell [2011] introduces cutting-edge techniques to approximate the cost or reward function of a generic MDP.

Rather than using the classical techniques to solve the optimization problem on the right-hand side of function 2.1, Puterman [1994] develops a linear programming approach. The number of variables in Putterman's dual problem is $(C+1)^n$ and the number of constraints is $B(n+1) \times C^n$, where C is the store capacity and $B(n)$ is the number of all partitions of all subsets of set N with $|N| = n$ members. Due to such enormous number of variables and constraints, neither column generation nor row generation algorithms could be efficient to tackle the problem.

By drawing inspiration from the primal LP formulation (P_0) and its dual problem (D_0) from Puterman [1994], Adelman [2004] formulates and interprets two primal-dual pairs of linear programs that are progressively more tractable. The first approximation of the dual problem is based on estimating the objective function value corresponding to the best decision when the state of the system is $x = (x_1, \cdots, x_n)$ by the sum of the objective function values when the state of the system in each customer (x_i) is independently dealt with. In other words, $V(x) = \sum_i V(x_i)$. The primal-dual problems of the first approximation are called P_1 and D_1, respectively. This is what Minkoff [1993] does, too. Such an approximation restricts the feasible region of P_0, while it relaxes the feasible region of D_0. Hence, D_1 provides an upper bound on D_0's maximization problem. The number of decision variables in D_1 decreases to nC but the number of constraints is unchanged. Row generation is the only practical approach to solve D_1. To mitigate the computational difficulties due to having so many constraints in D_1, Adelman [2004] introduces an additively separable approximation to the routing costs R, which further relaxes D_1. As a result, optimizing D_2 provides another upper bound for the original problem D_0. Adelman [2004] also shows that any feasible solution in either D_1 or D_2 provides a lower bound for P_0 and the optimal solution to D_1 provides the tighter lower bound. However, solving D_2 is easier compared to D_1. The lower bound obtained by solving D_1 or D_2 is useful to evaluate any algorithm solving P_0 approximately. The method to calculate the bounds is applicable to multiple products. However, as each customer-product is considered as a node, split deliveries of different products to each customer are inevitable.

Jaillet et al. [2002] define a SIRP involving a central depot as well as various satellite facilities which the drivers can visit during their shift to refill their vehicles. In case of a stockout, a direct delivery is made and a penalty cost is incurred. The authors assume that the central supplier has no reliable monitoring of local inventories except at the time of a delivery. This rules out any strategies that would schedule deliveries based on the current inventory levels. Jaillet et al. [2002] reduce the problem from an annual time base to a biweekly rolling planning period via the approximations of delivery costs.

Kleywegt et al. [2004] formulate an MDP model of the SIRP and propose approximation methods to find good solutions in reasonable computational time. In an earlier paper [Kleywegt et al., 2002], the authors formulated the SIRP with direct deliveries, i.e., one delivery per trip, as an MDP and proposed an approximate dynamic programming approach for its solution. Kleywegt et al. [2004] extend both the formulation and the approach to handle multiple deliveries per trip. They present a solution approach that uses decomposition and optimization to approximate the value function. Specifically, the overall problem is decomposed into smaller subproblems, each designed to have two properties: (a) it provides an accurate representation of a portion of the overall problem, and (b) it is relatively easy to solve. In addition, an optimization problem is defined to combine the solutions of the subproblems in such a way that the value of a given state of the process is approximated by the optimal value of the optimization problem.

Yu et al. [2012] aim to solve a large scale SIRP with split delivery. Service level of customers is considered, and unsatisfied customer demand in each period is backlogged. The number of routes each vehicle can perform in each period is not limited. Yu et al. [2012] propose an approximate model, which significantly reduces the number of decision variables compared to its corresponding exact model. They then develop a hybrid approach that combines the linearization of nonlinear constraints, the decomposition of the model into sub-models with Lagrangian relaxation, and a partial linearization approach for a sub-model. Table 2.4 shows the most important SIRP papers we have found in the literature. All the works assume a single product and an unlimited production supply.

2.6 Inventory control of perishables

Deterioration refers to damage, spoilage, dryness, or vaporization of products [Goyal and Giri, 2001]. Deteriorating products are divided into two categories: (a) perishable products, e.g. vegetables, meats, human blood, having a maximum usable life time, and (b) decaying products, e.g. alcohol and gasoline, having an unlimited shelf life but decreasing in quantity due to vaporization. We only consider perishables in our discussion. In the inventory modeling of perishables, a total profit maximization or total cost minimization approach is predominantly pursued [Minner and Transchel, 2010]. The usual costs include ordering, holding, outdate, and shortage costs. In the model settings, inventory system is reviewed periodically or continuously, products are single or multiple, demands are deterministic or stochastic, shelf life is deterministic or stochastic, replenishment lead time is zero or positive, shortages are backlogged or lost sale, and issuing policy is first-in-first-out (FIFO) or last-in-first-out (LIFO). Customer capacity and service level could be part of side constraints.

Table 2.4: Stochastic inventory routing models and solution methods: stochastic demand, single product, non-periodic delivery, unlimited supply

Study	Number of vehicles	Repleni. policy	Direct delivery	Plann. horizon	Shortage	Other characteristics	Solution method
Jaillet et al. [2002]	unlimited	OU with constant replenishment interval	only when stockouts happen	infinite; roling horizon	N/A	actual demands are realized at the time of a delivery; stock-outs are compensated through urgent direct deliveries at a cost	heuristic; approximating delivery costs using cost-to-serve estimation for each customer
Adelman [2004]	unlimited	ML	no	infinite	lost sale		LP formulations for SDP; approximations of primal and dual problems; providing bound for the LP
Minkoff [1993]	unlimited	ML	no	infinite	lost sale		MDP; customer decomposition
Berman and Larson [2001]	unlimited	ML	no	finite		demands are realized upon arrival	SDP
Kleywegt et al. [2002]	K	ML	yes	infinite	lost sale	direct deliveries	MDP; iterative heuristic; approximate dynamic programming
Kleywegt et al. [2004]	K	ML	no	infinite	lost sale	the approach in Kleywegt et al. [2002] is extended to incorporate multiple deliveries per trip	MDP; iterative heuristic; decomposition; approximate the value function
Yu et al. [2012]	unlimited	ML	no	finite	backlogging	split deliveries; multiple-tours per period by each vehicle; service level	approximate model providing a LB; transform the approximated model into a deterministic one; Lagrangean relaxation to decompose the deterministic model
Coelho et al. [2014b]	single	OU and ML	when stockout happens	finite	N/A	outsourced direct deliveries; emergency transshipments to avoid stockouts; service level to determine trigger point	point estimates of future demands; sequential decision making; (R,s,S) policy; heuristic algorithms; proactive and reactive policies; ALNS

Perishable products constitute the majority of sales revenue of the food retail industry. Perishables accounted for 52% of the 2011 total supermarket sales of about $459 billion in the US, and hence mismanagement of perishable products represents a major threat to the profitability of companies in the food retail industry. Roughly 10% of all perishable goods go to waste before consumers purchase them [Kouki and Jouini, 2015a]. A survey by the National Supermarket Research Group reported an average loss of $34 million a year due to spoilage in one major 300-store grocery chain in the US. Thus, finding effective inventory management policies for perishable products has always been of great interest to both practitioners and academic researchers [Chao et al., 2015].

On the one hand, the increasing food prices and at the same time billions of dollars' worth of food expiring every month raise major concerns in public discussion. On the other hand, supermarkets lose revenue when products are not available on the shelf. Especially in food retail distribution, some of the strongest requirements for practical use of inventory management systems are safety stock planning approaches that can deal with the demand seasonality and can satisfy service level requirements. This issue is identified as one of the most important key performance indicators in fresh food industries [Minner and Transchel, 2010]. The worldwide average out-of-stock rate is 8.3%; it accounts for 8.6% in Europe, and 7.9% in the US [Minner and Transchel, 2010].

It is widely known and accepted that stockout penalty cost approaches are difficult to implement and hardly used in practice, mainly due to the estimation of their value, e.g., to the quantification of the loss of goodwill. In fresh food retail, this is particularly accentuated by cross selling arguments, i.e., the unavailability of fresh food products impacts the sales of other product categories and even mid-term and long-term outlet choice. Therefore, a service-level approach becomes essential for fresh food inventory management [Minner and Transchel, 2010]. Here, we investigate inventory control of perishables from the RMI and the VMI points of view.

2.6.1 Inventory control of perishables in an RMI system

Recall that in an RMI system, the stores decide on when and how much to order, independently of each other. Therefore, the main decision variables in an RMI perishable inventory system are the order time and the order quantity. In order to place an order, the current inventory level and age of the stocked products (state of the system) are observed. In most of the problem settings, an MDP provides an exact solution approach. However, the computation of the optimal order for every state of the system using the well-known techniques such as (stochastic) dynamic programming for a finite horizon or value function for an infinite horizon is in general intractable because of the curse of dimensionality. Thus, many researchers turned to seek effective heuristic policies for these problems, and almost all heuristics developed so far have been focused on independent and identically distributed demands [Chao et al., 2015]. The ordering policies could be divided into: (a) periodic-review such as (R,S), (R,Q), (R,s,S), (R,s,Q), and (b) continuous-review such as (s,S), (s,Q). In the periodic-review policies, R refers to the number of periods between two consecutive reviews of the inventory system. In both types of reviews, s denotes the inventory level triggering an order, whereas S and Q are order-up-to level and order-up-to quantity values, respectively. A brief definition of each policy is provided hereunder.

-(R,S) : every R periods order up to level S,
-(R,Q) : every R periods order Q units,
-(R,s,S) : every R periods review the inventory position, and order up to level S if the inventory position is below s,
-(R,s,Q) : every R periods review the inventory position, and order Q units if the inventory position is below s,
-(s,S) : whenever inventory position hits s order up to level S,
-(s,Q) : whenever inventory position hits s order Q units.

Note that base stock policies $(R,S-1,S)$ and $(S-1,S)$ are special cases of the periodic-review (R,s,S) policy and the continuous-review (s,S) policy, respectively, where $s=S-1$. The $(R,S-1,S)$ and $(S-1,S)$ policies are often used for slow-moving expensive products where ordering cost is negligible compared to the price of the product. In our literature review, we confine ourselves to periodic-review policies for a single fast-moving product with stochastic demands and a deterministic shelf life. The most widely used periodic-review ordering policies are (R,S) [Chiu, 1995; Cooper, 2001; Deniz et al., 2010] and (R,s,S) [Broekmeulen and Van Donselaar, 2009; Lian and Liu, 1999]. For other cases, the reader is referred to the following works on perishable products:

(a) continuous-review [Berk and Gurler, 2008; Kouki et al., 2015b],
(b) multiple products [Nahmias, 2011; Karaesmen et al., 2011],
(c) deterministic demands [Hsieh and Dye, 2010; Hsu, 2000],
(d) stochastic shelf life [Nahmias, 2011; Kouki and Jouini, 2015a; Kouki et al., 2014],
(e) joint inventory control and pricing [Burnetas and Smith, 2000; Chen et al., 2014; Li et al., 2009; Chen and Sepra, 2013],
(f) slow-moving products [Nandakumar and Morton, 1993; Olsson and Tydesjo, 2010].

When demands are stochastic, obtaining optimal parameters in periodic-review policies even for a single perishable product with deterministic shelf life is notoriously complicated. The fixed shelf life perishability problem remains a complex problem when the product lifetime is longer than two units of time in a periodic review system [Kouki and Jouini, 2015a]. Hence, researchers have worked on approximating outdate costs [Broekmeulen and Van Donselaar, 2009; Chiu, 1995] or calculating upper and lower bounds on the number of outdates [Cooper, 2001; Chiu, 1995]. There are models dealing with batch demands [Lian and Liu, 1999] or batch orders [Broekmeulen and Van Donselaar, 2009]. Finally, service level is regarded as a constraint in some papers including Minner and Transchel [2010], Adachi et al. [1999], and Broekmeulen and Van Donselaar [2009].

The interested reader is referred to Nahmias [2011], Karaesmen et al. [2011], and Goyal and Giri [2001] for the best review works on perishables. Table 2.5 shows the most interesting papers we have found on inventory control of perishables in RMI systems dealing with a single product, deterministic shelf life, stochastic demand, periodic review, and infinite planning horizon.

2.6.2 Inventory control of perishables in a VMI system

Most of the IRP and SIRP models in the literature assume an unlimited product shelf life. This is one of the main obstacles for the application of the IRP and SIRP models in food logistics management. Inventory control of perishables in a system where inventory and routing-related decisions are made centrally for products with limited

Table 2.5: Inventory control of perishables in an RMI models and solution methods: single product, stochastic demand, infinite planning horizon, periodic review, limited deterministic shelf life

Study	Demand	Repleni. lead time	Issuing policy	Shortage	Other characteristics	Solution method
Lian and Liu [1999]	discrete	zero	FIFO	back-order	batch demands	$(R=1,s,S)$ policy
Minner and Transchel [2010]	discrete	positive	FIFO and LIFO	lost sale	service level constraints; nonstationary demand distribution	dynamic $(R=1,Q)$ policy
Chiu [1995]	continuous	positive	FIFO	back-order		(R,S) policy
Cooper [2001]	discrete	positive	FIFO	lost sale		$(R=1,S)$ policy; upper and lower bounds on outdates per period
Deniz et al. [2010]	discrete or continuous	zero	FIFO, LIFO, and hybrid	lost sale	two-period shelf life; joint demand distribution for different ages; demands for old (new) can be satisfied from new (old); age-dependent selling price	$(R=1,S)$ and $(R=1,Q)$ policies
Chao et al. [2015]	discrete or continuous	zero	FIFO	back-order and lost sale	nonstationary and correlated demands over planning horizon	approximation algorithms that guarantee worst-case performance
Adachi et al. [1999]	discrete	positive	FIFO	back-order	minimum stock level to be respected; age-dependent selling price; nonstationary demand distribution	MDP
Broekmeulen and Van Donselaar [2009]	discrete or continuous	positive	FIFO and LIFO	lost sale	batch ordering; safety stock; nonstationary demand distribution	(R,s,nQ) policy

shelf lives is a real problem faced by almost all food retail chains. The size of such problems is very large in many applications. One of the typical retail chains in a country as small as Belgium dispatches over 18,000 perishable and dry products (except beverages) from a central warehouse to more than 800 stores throughout the country. Since perishable products in a food retail chain need to be kept and carried by special vehicles in a cool temperature, we can isolate their IRP or SIRP model from that of dry products. Though the real case usually consists of multiple perishable products, which are simultaneously shipped together, the main body of the existing research develops solution methods for a single perishable product. Profitability of the food retail industry highly depends on efficient handling of the products, as the profit margin of this sector is around 2%. Therefore, it is crucial to devise effective inventory routing policies to yield a competitive profit margin.

The problem statement of an IRP for perishables is exactly the same as that for non-perishables, except that products have a limited shelf life after which they have no value. In the deterministic case, shortages are not allowed. Moreover, thanks to the complete knowledge about the demands, nothing is deteriorated. This implies that the objective function in the IRP for perishables remains the same as what we have defined for non-perishables, i.e, minimizing total inventory holding costs and routing costs during a planning horizon.

The largest difference between the classical IRP and the IRP for perishables is the way in which the maximum delivery quantities to stores are defined. While the maximum delivery quantity in the IRP depends only on the physical storage capacity and the on-hand inventory at the customer's site, in the IRP for perishables it is restricted by not only these two parameters but also the maximum shelf life of the product.

In the IRP for perishables, delivery frequency plays a big role. Less frequent deliveries reduce the routing costs, but leave behind products with shorter remaining shelf lives in the following periods subject to not only holding cost and deterioration, but also undesirable freshness from the end customers' perspective. Moreover, the demand for the products may reversely be affected by the age of the inventory. If deliveries are executed more frequently, freshness of products and consequently customer satisfaction increases. However, higher routing costs are imposed to the system. Therefore, finding a right trade-off between costs and freshness is crucial. The main objective in most applications is minimizing the costs (or maximizing the profit), while freshness is controlled by imposing additional side constraints on delivery quantities.

Hemmelmayr et al. [2009b] investigate a problem on delivery of blood products from a blood bank to hospitals. Blood products are delivered to the hospitals using a small fleet of identical vehicles. As the size of a bag with a blood product is very small, vehicle capacity is ignored, though the route length has to be restricted. The objective is to minimize total routing costs during a finite planning horizon; no inventory costs are taken into account. As stock-outs may result in loss of life, hospitals actually prefer to have high inventory, even if this results in higher costs. Outdate costs need not be considered either, because their approach does not allow spoilage of blood products. They develop and evaluate two delivery strategies. The first strategy retains the concept of regions and the use of fixed routes, but uses integer programming techniques to optimally decide on delivery days. The integer programming-based approach employs a scheme in which the set of hospitals is divided into four regions and the hospitals in each region are served by a single vehicle using a fixed route, which simply skips those hospitals that do not require a delivery. At the heart of the integer programming model is the observation that shortcutting of a fixed route allows for the consideration of a substantial number of routes and may therefore provide adequate flexibility to achieve substantially reduced delivery costs. The second approach is based on viewing the problem as a PVRP with route length constraints but without capacity constraints. The challenge is to simultaneously select a visit combination for each customer and to solve the implied daily vehicle routing problems. As different visit frequencies may lead to feasible delivery patterns for hospitals, i.e., delivery patterns that do not result in product shortages and product spoilage, they allow each hospital to have a set of visit frequencies and thus of associated (periodic) visit combinations. Visit combinations that lead to an infeasible delivery pattern will be deleted. They develop a variable neighborhood search (VNS) algorithm to solve this variant of the PVRP.

Coelho and Laporte [2014c] consider an IRP for perishables, where inventory holding cost and selling price are age-dependent. The supplier has the choice to deliver fresh or aged products, and each case yields different holding costs and different revenues. The objective function maximizes the total sales revenue, minus inventory and routing costs. It is up to the retailer to decide which items to offer to customers, which will influence the associated revenue. Three different selling priority policies are investigated: (1) fresh first policy (FF), (2) old first policy (OF), (3) optimized priority policy (OP). The latter policy lets the model determine which items to sell at any given time period in order to maximize profit. This means that depending on the parameter settings, one may prefer to spoil some items and sell fresher ones because they generate higher revenues. Although they are similar, FF and OF policies are different from the traditional FIFO and LIFO policies common in inventory management. Under a FIFO policy, the first product delivered will be the first to be sold. This coincides with an OF policy only if deliveries from the supplier to the retailer is always of fresh items. However, when the supplier delivers products of different ages in different periods, the sequence of deliveries does not necessarily coincide with the ages of the products in inventory. They formulate this IRP for perishables as an MILP and devise an exact branch-and-cut algorithm for the solution of the various models.

Le et al. [2013] propose a mathematical model for the IRP for perishables using the concept of a feasible route. A feasible route is a route that starts from the depot, visits a subset of customers at most one time and then returns to the depot. This is different from the popular notion of a feasible route in the VRP, where a feasible route is defined as a route for which the sum of demands of customers on the route are less than the vehicle capacity. Le et al. [2013] use a column-generation based heuristic algorithm to solve the problem.

Al Shamsi et al. [2014] extend the classical IRP for perishables to include the cost of CO2 emissions due to transportation. Their model is similar to the model developed in Le et al. [2013] but with cost of CO2 emissions incorporated into the objective function along with transportation costs and inventory holding costs. The CO2 emissions are calculated based on the vehicle load and distance. The resultant model is a Mixed Integer Nonlinear Programming (MINLP) problem which is solved using a commercial solver.

Mirzaei and Seifi [2015] formulate an IRP for perishables in which the end customers' demand is a decreasing function of inventory age so that a portion of the demand is considered as lost sale if inventory is not as fresh as it could be. The resultant model is an MINLP but is linearized by the authors. The objective function is to minimize the total cost of routing, lost sale, and holding inventories. The mathematical model is solved up to optimality for small to medium size problems. The authors develop a hybrid simulated annealing (SA) and tabu search (TS) heuristic for solving larger problem instances. Table 2.6 shows the papers on the IRP for perishables that we have found in the literature.

2.7 Conclusions

In this chapter, we have reviewed several problem formulations and solution methods that are closely related to those studies in the thesis. The SIRP is a well-known problem in the literature. However, adding perishability of products and stochasticity of demands to the features of the problem complicates it significantly. As a result, applying the existing solution methods is mathematically very difficult and computationally very inefficient. In Chapter 3, we study the SIRP for perishables and develop different approximate solution methods to solve the problem. In the most efficient solution method we have developed to solve the SIRP for perishables, we need to solve a two-period VRP. In Chapter 3, a Matheuristic is developed to solve it, but the problem is further investigated in Chapters 4 and 5. More specifically, Chapter 4 introduces the two-period VRP as a subproblem of the most efficient solution method we have developed to solve the SRIP for perishables. The two-period VRP is also motivated as a chunk of a special DMPVRP to be solved in a rolling horizon. In this sense, it is discussed in Chapter 4 that the two-period VRP is neither a MPVRP, nor a PVRP, nor an IRP. Therefore, the common solution methods to solve the latter problems cannot be applied to solve the two-period VRP. Chapter 5 deals with the same two-period VRP under slightly different assumptions, which are less restrictive. Chapters 4 and 5 develop exact solution methods to solve the two-period VRP.

Table 2.6: Inventory routing models for perishables and solution methods: deterministic demand, finite planning horizon, single product, no shortage, limited deterministic shelf life

Study	Number of vehicles	Repleni. policy	Direct delivery	Periodic delivery	Production supply	Other characteristics	Solution method
Coelho and Laporte [2014c]	K	ML	no	no	limited	fresh first (FF) and old first (OF) issuing policies; different ages are delivered; extendable to include decaying products; age-dependent holding costs and selling prices	branch-and-cut
Le et al. [2013]	K	ML	no	no	unlimited		route-based formulation; column generation-based heuristic
Al Shamsi et al. [2014]	K	ML	no	no	unlimited	builds on the model by Le 2013; CO2 emission costs are considered	route-based formulation; simulation; Commercial solver
Mirzaei and Seifi [2015]	K	ML	no	no	unlimited	demands is a decreasing function of inventory age	arc-flow formulation; Hybrid SA-TS
Hemmelmayr et al. [2009b]	single	ML	no	no	unlimited	uncapacitated vehicle with a maximum length	arc-flow IP formulation; VNS

Chapter 3

Stochastic inventory routing for perishable products

3.1 Introduction

Consider a retail chain whose main goal is to optimize the long-term profit of a distribution network where products are shipped from a central warehouse (depot) to several stores. The decisions to be made are: (1) how much to deliver to each store in each period, and (2) which routes to use. If demands from the end customers to the stores are deterministic and known for the entire planning horizon, then this decision problem is known as the Inventory Routing Problem (IRP). Andersson et al. [2010] and Coelho et al. [2014a] provide excellent reviews of the IRP from the application and the methodological points of view, respectively.

In this chapter, we consider the IRP for a perishable product (e.g., a dairy product, flowers, fruits, or vegetables) with stochastic demands from the end customers. We develop four solution methods to tackle this problem: (1) an expected-value method in which stochastic demands are replaced by their expected values, (2) a deliver-up-to-level policy taking into account a high target service level, (3) a decomposition method in which an independent inventory control model is developed for each store while taking into consideration an estimation of the routing costs imposed by the store, (4) a decomposition-integration method which improves the solution obtained by the decomposition method through further analysis of the routing costs.

3.1.1 Motivation

In an IRP, the delivery quantities as well as the routes used to serve the stores are determined by a centralized decision-maker for the entire planning horizon. IRPs are generally very difficult to solve to optimality even when the distribution network is far smaller than those encountered in practice. One of the typical retail chains in Belgium (a rather small country) dispatches over 18,000 perishable and dry products (except beverages) from a central warehouse to more than 800 stores throughout the country. In contrast, as reported in the literature [Coelho and Laporte, 2014c], IRPs for a single product and with deterministic demands can rarely be solved to optimality when the number of stores is larger than 30.

In most real-life IRP applications, demands to stores are uncertain, thus giving rise to a more complex stochastic version of the IRP, denoted SIRP. In this stochastic setting, the value of the current inventory levels must be periodically transferred from the stores to the central warehouse. Taking this information into consideration, as well as whatever information is available about future demand, the central decision-maker determines the delivery quantities and the routes for only one period (or a few periods) ahead and implements these short-term decisions. Then, the actual demands from the end customers to the stores are observed, new inventory levels are reported to the warehouse, and new one-period (or short-term) decisions are made. Due to the complexity of such an integrated system, retail chains frequently rely on a sequential decision-making process, whereby each store independently uses its own inventory management system and places its order by neglecting the routing costs imposed to the retail chain. Then, vehicle routing models are used by the central office to determine the optimal delivery routes. Such a decision-making process does not necessarily yield an optimal profit for the complete retail chain, but provides a pragmatic approach for this complex system.

The common assumption of unlimited product shelf life is not applicable to perishables. This is one of the main obstacles for the application of the classical SIRP models in food logistics management. Perishable products constitute over 52% of sales revenue of the grocery retail chains [Chao et al., 2015], but roughly 10% of them go to waste before they are sold [Kouki et al., 2015b], while the profit margin in food retail industry hardly exceeds 2%; see Euro Bank [2009]; FMI [2014]; NAICS [2012]. Therefore, the profitability of this sector highly depends on efficient management methods, and it is crucial to devise effective inventory routing policies to ensure a competitive profit margin. The SIRP for perishables, or PSIRP, is the topic of this chapter. Note that, since perishable products need to be carried by special refrigerated vehicles, food retail chains can isolate their SIRP model from their model for dry products.

In a PSIRP, the delivery frequency plays a major role in determining the profit, service level, and freshness. Indeed, infrequent deliveries with large delivery quantities reduce the routing costs, decrease the risk of facing lost sales, and increase service level. However, they result in units with shorter remaining shelf lives and subject to deterioration in the following periods, due to the stochasticity of demand. Conversely, if smaller quantities are delivered more frequently, then the freshness of products – and consequently, customer satisfaction – increases, and deterioration costs decrease. However, such a policy imposes higher routing costs to the system and may bring about more lost sales. Our main objective is to maximize the profit; a predefined minimum service level is considered as a hard constraint to be respected and freshness is regarded as a lateral consequence.

3.1.2 Scientific contributions

The main contributions of this chapter are summarized here:

- We propose different solution methods to solve the PSIRP, a problem scarcely explored in the literature despite its wide applicability. Each method emphasizes some, though not necessarily all, crucial features of the real logistics problem, such as perishability of the product, stochasticity of the demand, or service level constraints. Each method may yield benefits over the other ones in terms of total profit, average freshness, or simplicity of implementation.

- Based on extensive computational tests, we examine the performance of each solution method and the influence of different parameter settings.

- Managerial insights are drawn by analyzing the impact of store capacity and shelf life on the expected profit. Moreover, we show that a simple replenishment policy can be derived from a more complex solution method, while yielding similar efficacy.

3.1.3 Related works

First of all, note that extending the SIRP model introduced in the transition function 2.1 to perishables is extremely difficult. Indeed, in the presence of perishability, the transition function is far more complicated, and this complexity increases as the shelf life increases. Moreover, transition function 2.1 is applicable when the probability function of the demand is stationary over the planning horizon, i.e., the demand of end customers has the same distribution function with the same mean and variance. It can be easily verified by other research works, see Van Donselaar et al. [2010], that the demand of end customers food retail stores varies significantly from day to day.

Research papers on the SIRP for perishables are scarce in the literature. We have found only two research papers in this category. Hemmelmayr et al. [2010] consider delivery of blood products with stochastic demands as an extension of Hemmelmayr et al. [2009b] where demands are deterministic. They use sampling to solve their problem, where R random realizations of the demands for the entire planning horizon are generated. Then, a two-stage approach is followed. In the first phase, based on a sample of size R, i.e., R deterministic demand scenarios for the entire planning horizon, they formulate an IP that considers a single uncapacitated vehicle with a maximum route duration. To balance delivery costs and waste costs, they choose to limit the probability that spoilage occurs to at most 5%. In other words, they sample product usage during the spoilage period and take the 5% quartile as the maximum inventory level at the hospital (spoilage capacity). Vertices are visited in such a way that under any of the realizations none of the vertices faces a stock-out nor does its inventory go beyond the spoilage capacity. For the routing component, the IP model relies on shortcutting a pre-determined fixed route. It implies that a TSP solution covering all vertices is at hand. This solution is exploited to construct another TSP solution where a subset of vertices are served. In the solution, they simply cut arc (i, j) if either node i or node j is not served in the corresponding period. The remaining partial paths are connected to make a complete TSP solution. The second phase suggests emergency daily deliveries to avoid stock-outs. Further constraints are incorporated into the initial IP formulation and the objective function is extended in order to include the emergency deliveries. To this end, some recovery or recourse mechanisms are created to handle shortages if they occur. The recourse actions are (1) changing the quantities of planned deliveries, (2) introducing out-and-back emergency deliveries (direct deliveries) to hospitals that are likely to experience a shortage based on the inventory at the beginning of the day, (3) introducing a single emergency delivery route for each day visiting all hospitals that are likely to run out of product on that day, and (4) introducing emergency deliveries into the regularly planned delivery routes. The objective is to minimize total travel distance during the planning period and the costs associated to the recourse actions. Variable Neighborhood Search (VNS) is exploited to solve the two-phase IP formulation.

A main obstacle to adopting the solution approach of Hemmelmayr et al. [2010] is that the vehicles in their model are uncapacitated, whereas we consider capacitated vehicles. Moreover, the assumption that TSP routes can be pre-determined is extremely restrictive. Finally, for the solution approach developed by Hemmelmayr et al. [2010] to remain efficient, only a very limited number of realizations of demands can be considered, and this might not be sufficient to represent the universe of potential scenarios.

Soysal et al. [2015] consider a SIRP for perishables in food supply chain for a single product. Arc costs are vehicle-load dependent in order to reflect the fact that fuel consumption depends on the vehicle load. The vehicle-load dependency of arc costs also models CO2 emission and penalizes it. The authors use chance-constrained programming to formulate the problem. The objective function includes the expected inventory holding costs, waste costs, load-dependent arc costs, and driver costs. The constraints are: (1) inventory balance equations in terms of the expected demand, inventory, and waste, (2) routing-related constraints including flow conservation and sub-tour elimination, and (3) a service level constraint to ensure a high probability of not running out of stock in each period. The service level constraint is approximated by a linear inequality as a function of the expected waste and delivery quantity. The authors use a commercial MILP solver to solve the problem, followed by a simulation model to compare the solutions obtained from the model with its simplified versions where perishability or fuel consumption is not taken into account.

In the work of Soysal et al. [2015], all constraints are simplified from the very beginning, since the stochastic demands are replaced by their expected value in the constraints. Indeed, the probability distribution of demand is only considered in the service level constraint, which is equivalent to a constraint we impose in our model about the minimum delivery quantity. All other constraints, however, bear on the expected demand rather than the random demand variable. In this sense, their model could be regarded as a deterministic PIRP. Table 3.1 shows the two papers we have found on the SIRP for perishables as well as how our work is positioned with respect to them.

The remainder of this chapter is organized as follows. Section 3.2 presents the problem statement. Sections 3.3-3.7 propose four solution methods. In Section 3.8, we develop a matheuristic to solve the optimization problem arising in one of the solution methods. Section 3.9 presents a heuristic algorithm for the deterministic case where full information is available about future demands. An exhaustive computational study is carried out in Section 3.10. Results of the computational experiments are analyzed and discussed in Section 3.11, where we also draw algorithmic and managerial insights. Finally, concluding remarks are formulated in Section 3.12.

3.2 Problem statement

We consider a generic retail chain that attempts to maximize the expected net profit generated by the sales of a single perishable product. The net profit is measured by deducting acquisition, distribution, and other miscellaneous costs from the total revenue. Acquisition costs and revenue mostly depend on the quantities delivered to the stores, whereas distribution costs are also a consequence of the way the vehicles are dispatched. Miscellaneous costs are viewed as independent of either decision and are regarded as constant costs. As a result, we do not consider them in the objective func-

Table 3.1: Stochastic inventory routing for perishables models and solution methods: stochastic demand, single product, non-periodic delivery, unlimited production supply, limited deterministic shelf life

Study	Number of vehicles	Repleni. policy	Direct delivery	Planni. horizon	Shortage	Other characteristics	Solution method
Soysal et al. [2015]	K	ML	no	finite	lost sale	service level; penalty for waste; load-dependent arc cost (CO_2 emission)	arc-flow formulation; chance-constrained programming
Hemmelmayr et al. [2010]	unlimited	ML	when stockout happens	finite	N/A	service level; extension of Hemmelmayr et al. [2009b] (deterministic demands); uncapacitated vehicle with maximum lengths; emergency direct and route deliveries when stockout happens	pre-determined TSP solution; arc-flow IP formulation; VNS
Our SIRP for perishables	unlimited	ML	no	finite	lost sale	service level; FIFO, applicable for positive holding costs and decaying products	SDP; routing/inventory decomposition-integration; Matheuristic

tion: the net profit is simply computed as Revenue – Acquisition costs – Distribution costs, and its expected value defines the objective function.

We assume a finite planning horizon of length T. We consider an implicit complete graph $G = (V,A)$, whose vertices represent the depot and the stores, and arcs represent the road segments between pairs of vertices. The distance and/or travel time from vertex i to vertex j is denoted as c_{ij}. Products are picked up from the depot and delivered to the stores. Each route starts from the depot, ends at it, and cannot exceed a predefined length. (Most of our work could actually handle different types of feasibility constraints on routes, but we limit ourselves to route length for simplicity.) The demand in period t from end customers to each store i is an integer random variable D_{ti} (assumed independent for all periods and all stores) with known probability distribution. We define L as the deterministic shelf life of each unit of product from the moment it is delivered to the stores. The acquisition cost of each unit is a. All units delivered in period t have the same selling price p during L periods, and unsold units perish at the end of period $t+L-1$ with no salvage value. Unmet demand leads to lost sales but does not generate any other cost. The inventory holding cost is considered to be zero. We assume that the depot holds an unlimited supply. Capacity of store i is denoted by C_i. The retail chain owns an unlimited number of identical vehicles with capacity Q. Each vehicle incurs a fixed cost K per period when it is used, and a variable cost equal to c_{ij} when traveling from vertex i to vertex j. Split deliveries are not allowed, i.e., each store is served by at most one vehicle in each period.

The retail chain uses a centralized decision-making system to determine delivery quantities and routes in each period. The inventory state in store i at the beginning of period t is denoted by $X_{ti} = (x_1, x_2, \ldots, x_{L-1})_{ti}$, where $(x_k)_{ti}$ is the inventory level with remaining shelf life k. At the beginning of each period t, based on the inventory states, the retail chain decides about the delivery quantities, y_{ti}, and the routes, R_t, to be used in the current period. There is no time window for the delivery to stores, but in each period each vehicle is allowed to perform at most one route with a predefined maximum length. The delivery lead time is zero, i.e., the delivery quantities y_{ti} are available on the shelves at the beginning of period t, right after the decision is made by the retail chain. This is not an unrealistic assumption as in practice decisions are made at the end of period $t-1$ and the vehicles are dispatched over night. The real demand for period t is observed during the period and after quantity y_{ti} has been delivered. We assume that the oldest units of product are sold first (FIFO issuing policy), i.e., $(x_k)_{ti}$ is stored until $(x_{k-1})_{ti}$ is used up or perished.

A hard constraint on the inventory control model is that a predefined Target Service Level (TSL) must be respected in every period and in every store. More precisely, the total inventory available in store i at the beginning of period t after y_{ti} is delivered, must be such that the probability of not incurring a stockout in period t is at least equal to TSL. This constraint is enforced in most solution methods we develop in order to ensure fair comparisons. In practice, the average service level of perishables can be estimated to be around 92% in Europe and in the USA, as cited by Minner and Transchel [2010].

In the following four sections, we develop four different solution methods to solve the PSIRP introduced in this section. The corresponding algorithms appear at the end of each section. In the following sections, we will use the notations summarized in Table 3.2.

Table 3.2: Indices, parameters, and decision variables

Indices:	
i, j	indices for vertices (depot and stores)
k	index for remaining shelf life
r	index for routes
t	index for period
Parameters:	
T	length of the planning horizon
N	number of stores
L	shelf life of the product
TSL	target service level to be respected in every period and in each store
a	acquisition price of each unit of the product
p	selling price of each unit of the product
C_i	capacity of store i
$(x_k)_{ti}$	inventory level with remaining shelf life k in period t before delivery in store i
$(x_1, x_2, \ldots, x_{L-1})_{ti}$	state of the system in period t in store i before delivery
I_{ti}	total inventory at the beginning of period t in store i before delivery
D_{ti}	random demand of end customers in period t in store i (integer-valued)
$Pr(D_{ti} = d)$	probability function of demand in period t in store i
Q	capacity of each vehicle
c_{ij}	distance and travel cost from vertex i to vertex j
F_{ti}	estimated cost-to-serve assigned in period t to store i
Decision variables:	
y_{ti}	delivery quantity in period t to store i (integer values)
R_t	set of routes used in period t (index r)
π	expected profit generated by all stores over the planning horizon

3.3 The expected value method

A classical way to reduce the complexity of stochastic models is to replace random variables by their expected values. Following this approach, the *expected value method* (EV) considers that demands are deterministic and equal to $E(D_{ti})$, for all i,t. In order to describe this method more precisely, let us again assume that the inventory levels in period t in store i before delivery are given as $(x_1, x_2, \ldots, x_{L-1})_{ti}$, so that the total inventory level is:

$$I_{ti} = \sum_{k=1}^{L-1} (x_k)_{ti}. \tag{3.1}$$

At the beginning of period t, for each store i, EV_λ determines the delivery quantity as follows: if the current inventory level I_{ti} is larger than or equal to the average demand $E(D_{ti})$, the delivery quantity is zero; otherwise, EV_λ delivers enough to satisfy the expected demands of λ periods ($1 \leq \lambda \leq L$), including the current period, provided that store capacity C_i is respected. Then, a VRP is solved based on these delivery quantities, real demands are observed, new inventory levels are calculated for the next period, and the same decision making process is repeated in period $t+1$.

All necessary computations can be carried out as follows. Assume that in period t, y_{ti} units are delivered and the actual demand d_{ti} is observed in store i. Then, the inventory level of the store in period $t+1$ is determined by relation (3.3), where $(z)^+ = \max(z,0)$ and $(x_L)_{ti} = y_{ti}$ by convention:

$$((x_1,x_2,\ldots,x_{L-1})_{ti}, y_{ti}) \xrightarrow{d_{ti}} (x_1,x_2,\ldots,x_{L-1})_{t+1,i} \tag{3.2}$$

$$(x_k)_{t+1,i} = ((x_{k+1})_{ti} - (d_{ti} - \sum_{l=1}^{k}(x_l)_{ti})^+)^+, \quad \text{for } k = 1,\ldots,L-1. \tag{3.3}$$

The expected value algorithm (EV_λ)
Begin

Step 0. Set $t = 1$.

Step 1. For each store i, if $I_{ti} \geq E(D_{ti})$, set $y_{ti} = 0$; otherwise, set $y_{ti} = \min\{C_i - I_{ti},\ \lfloor E(D_{ti}) + \ldots + E(D_{t+\lambda-1,i}) \rfloor - I_{ti}\}$.

Step 2. Solve a VRP for the delivery quantities y_{ti}'s and serve the stores through the optimal VRP routes.

Step 3. For each store i, observe the actual demand in period t, say d_{ti}. Calculate the state of the system in period $t+1$, i.e., $X_{t+1,i}$ by Relations (3.3). Set $t = t+1$ and go to Step 1.

End

At each store, the expected value method can be viewed as an (R,s,S) policy, where $R = 1$, $s = E(D_{ti})$, and $S = \min\{C_i,\ \lfloor E(D_{ti}) + \ldots + E(D_{t+\lambda-1,i}) \rfloor\}$. Note that EV_λ does not enforce the target service level TSL in *every* period, as it does not access demand probability function. Consider a scenario where $I_{ti} \geq E(D_{ti})$ but I_{ti} is very close to $E(D_{ti})$. Based on the EV_λ method, nothing is delivered in period t to store i, but there is a remarkable chance that store i encounters a stockout in period t, as there is very little buffer inventory against excess demand beyond $E(D_{ti})$. This phenomenon happens in the periods right before the delivery periods.

However, by choosing a large value of λ, a buffer inventory against excess demand is provided in more periods, and therefore we can ensure the TSL in more periods. $\lambda = L$ creates the biggest number of such periods, and it is a suitable setting especially for small values of L. This compensates the likely poor service level in the periods right before the delivery periods so that a higher average service level is attained. In our computational experiments, we set $\lambda = L$, and EV_L is simply shown by EV. Not surprisingly, by this setting, the service level obtained with EV turns out to be high in our computational experiments, especially for larger values of L. In this chapter, we regard EV just as a basis to which other solution methods are compared.

3.4 A deliver-up-to-level method

As compared to EV, more effective solution methods would exploit the probability function of demand rather than merely its expected value. In particular, when the target service level TSL is high, a simple replenishment policy is to deliver the minimum number of units such that TSL is satisfied. Denote this quantity by $y_{ti}^{(1)}$, which is the smallest integer value of y_{ti} satisfying Inequality (3.4):

$$Pr(D_{ti} \leq y_{ti} + I_{ti}) \geq TSL. \tag{3.4}$$

Given the quantities $y_{ti}^{(1)}$, a VRP can be solved in order to determine the optimal delivery routes for period t. Then, real demands are observed and the process is repeated for period $t+1$. We denote by UL_1 this simple *deliver-up-to-level* policy which attempts to satisfy TSL for one period. A drawback of this policy is that, in the replenishment stage, it completely neglects the routing costs generated by the deliveries. In practice, if at least one unit of demand is observed in every period and in every store, then all stores are served in all periods. A somewhat moderated policy would be to deliver a bigger quantity to each store in such a way that on average, each store is visited every λ periods, with $\lambda \leq L$. In other words, the retail chain would deliver the smallest integer value y_{ti} satisfying Inequality (3.5):

$$Pr(D_{ti} + \cdots + D_{t+\lambda-1,i} \leq y_{ti} + I_{ti}) \geq TSL. \tag{3.5}$$

Denote by $y_{ti}^{(\lambda)}$ this quantity, which suffices to meet the demand of λ consecutive periods with probability TSL. If $y_{ti}^{(\lambda)} + I_{ti}$ is larger than the store capacity C_i, we reset $y_{ti}^{(\lambda)} := C_i - I_{ti}$. We call the corresponding method UL_λ.

The deliver-up-to-level algorithm (UL_λ)
Begin

Step 0. Set $t = 1$.

Step 1. For each store i, if $Pr(D_{ti} \leq I_{ti}) \geq TSL$, set $y_{ti} = 0$; otherwise, calculate the smallest integer value y_{ti} satisfying Inequality (3.5), i.e., $y_{ti}^{(\lambda)}$. If $y_{ti}^{(\lambda)} > C_i - I_{ti}$, set $y_{ti}^{(\lambda)} = C_i - I_{ti}$.

Step 2. Solve a VRP for the delivery quantities $y_{ti}^{(\lambda)}$ and serve the stores through the optimal VRP routes.

Step 3. For each store i, observe the actual demand in period t, say d_{ti}. Calculate the state of the system in period $t+1$, i.e., $X_{t+1,i}$, by Relations (3.3). Set $t = t+1$ and go to Step 1.

End

Note that, in this policy, we set $y_{ti} = 0$ whenever I_{ti} suffices to satisfy TSL in period t, since a positive delivery quantity would increase the routing costs in period t. If the inventory does not suffice to satisfy TSL in period t, we deliver $y_{ti}^{(\lambda)}$. In this sense, UL_λ can be viewed as an (R,s,S) policy where $s = q^{(1)}$, $S = \min\{C_i, q^{(\lambda)}\}$, and $q^{(\lambda)}$ is the smallest integer such that $Pr(D_{ti} + \cdots + D_{t+\lambda-1,i} \leq q^{(\lambda)}) \geq TSL$. As TSL must be respected in every period, we set $R = 1$. Note that $\lambda = 1$ tends to provide the freshest products on shelf thanks to daily delivery, whereas bigger values of λ yield lower routing costs and possibly higher profit.

3.5 A decomposition method

The deliver-up-to-level method UL_λ mostly focuses on the target service level in order to determine the delivery quantities, and downplays the importance of revenues and routing costs. Our next method relies on a Stochastic Dynamic Programming (SDP) model to account more explicitly for these elements .

In the SDP model, the state of the system in period t is defined by the inventory levels in all stores, i.e., $((x_1,\ldots,x_{L-1})_{t1},\ldots,(x_1,\ldots,x_{L-1})_{tN})$. The decision variables are the delivery quantities in period t, i.e., $(y_{t1},\ldots,y_{tN})$, and the routing decisions. Given a decision on delivery quantities in period t, the direct costs (acquisition and routing) as well as the expected revenue in period t can be formulated, as well as the potential states of the system in period $t+1$ and the transition probabilities. Theoretically, one can set SDP relations to determine the optimal delivery quantities in each period based on the state of the system. However, this can only be applied to very small-size instances. Considering N stores, a maximum shelf life L, and inventory levels to be integers in the interval $[0,C]$, there are $(C+1)^{N(L-1)}$ potential states in each period. Therefore, it is necessary to resort to heuristic methods to solve even small instances through SDP.

In our approach, we solve an independent SDP for each store, with the aim to optimize an estimate of the expected revenue generated by the store over the planning horizon. As a result, though such a decomposition yields sub-optimal solutions, the complexity of the problem no longer depends exponentially on the number of stores. With our previous notations, the number of states in each period for each store is $(C+1)^{(L-1)}$, which is computationally tractable for small values of L. In each period, the SDP relations allow us to determine a delivery quantity to each store, based on its current inventory level, while neglecting the routing costs. Then, we next solve a VRP to obtain optimal routes for these delivery quantities. We call this the *decomposition method* (*DE*).

Since the SDP model considers each store independently, it cannot properly account for the routing costs. Therefore, in the model associated with store i, we charge a fixed *cost-to-serve* F_{ti} if the store is visited in period t. This cost-to-serve acts as a surrogate for the routing cost generated by the delivery to store i. The choice of F_{ti} will be discussed later.

Let us use the shorthands $X_{ti} = \big((x_1)_{ti},\ldots,(x_{L-1})_{ti}\big)$, $X_t = (X_{t1},\ldots,X_{tN})$, and $Y_t = (y_{t1},\ldots,y_{tN})$, so that (X_t,Y_t) denotes the complete state of the system at time t after the quantities $y_{t1},\ldots,y_{tN}$ have been delivered. We define $f_{ti}(X_{ti},y_{ti})$ as the total expected profit for store i from period t until the end of the planning horizon when the state of the store is X_{ti} and the delivery quantity is y_{ti}. The function f_{ti} includes total revenue, acquisition costs, and the cost-to-serve. The optimal expected profit generated by store i from period t to the end of the horizon is denoted $f^*_{ti}(X_{ti})$, that is,

$$f^*_{ti}(X_{ti}) = \max_{y^{(1)}_{ti} \le y_{ti} \le C_i - I_{ti}} f_{ti}(X_{ti},y_{ti}), \tag{3.6}$$

where $y^{(1)}_{ti}$ is the smallest integer satisfying Inequality (3.4). The optimal delivery quantity is specified by Equation (3.7):

$$y^*_{ti} = y^*_{ti}(X_{ti}) = \arg\max_{y^{(1)}_{ti} \le y_{ti} \le C_i - I_{ti}} f_{ti}(X_{ti},y_{ti}). \tag{3.7}$$

In order to determine the optimal delivery quantity y^*_{ti}, we can solve the recursive Equations (3.8) by backward induction:

$$\begin{aligned} f_{ti}(X_{ti}, y_{ti}) &= -F_{ti} \cdot \mathbb{1}(y_{ti} > 0) - a y_{ti} \\ &+ Pr(D_{ti} > I_{ti} + y_{ti})\big(p(I_{ti} + y_{ti}) + f^*_{t+1,i}(0, \ldots, 0, 0)\big) \\ &+ \sum_{d=0}^{I_{ti}+y_{ti}} Pr(D_{ti} = d)\big(pd + f^*_{t+1,i}(X_{t+1,i})\big), \end{aligned} \tag{3.8}$$

where $X_{t+1,i} = (x_1, \ldots, x_{L-1})_{t+1,i}$ is defined by Equation (3.3).

The first term in Equation (3.8) is an estimate of the routing cost incurred to serve store i; the second term is the acquisition cost of y_{ti} units; the third term accounts for the expected revenue collected from store i when the demand in period t is larger than the inventory available at i in period t, and for the expected profit in periods $t+1, \ldots, T$; similarly, the last term expresses the expected revenue in period t and the expected profit in periods $t+1, \ldots, T$ when the demand does not exceed the available inventory. In order to solve (3.8), we use the boundary condition:

$$f^*_{T+1,i}((x_1, x_2, \ldots, x_{L-1})_{T+1,i}) = \frac{a}{2} I_{T+1,i}, \tag{3.9}$$

where the right-hand side of (3.9) is an estimate of the profit generated by the inventory left over at the end of the horizon. In our computations, T will be large enough that the effect of (3.9) will not be significant.

The decomposition algorithm *(DE)*
Begin

Step 0. Set a cost-to-serve, F_{ti}, for each store i and each period t based on one of the algorithms described in Section 3.6. Set $t = 1$.

Step 1. Use Equations (3.7)–(3.8) to determine a delivery quantity to each store i in period t, y^*_{ti}, given the state of the system $X_{ti} = (x_1, \ldots, x_{L-1})_{ti}$.

Step 2. Solve a VRP for the delivery quantities y^*_{ti} and serve the stores through the optimal VRP routes.

Step 3. For each store i, observe the actual demand in period t, say d_{ti}. Calculate the state of the system in period $t+1$, i.e., $X_{t+1,i}$ by Relations (3.3). Set $t = t+1$ and go to Step 1.

End

Note that the quantities y^*_{ti} determined by Equations (3.7)–(3.8) are not optimal for the original PSIRP but only for the decomposed problem. However, this approach might yield reasonably good solutions for the PSIRP provided that the costs-to-serve F_{ti} are reliable estimates of the actual routing costs. In Section 3.6, we introduce two methods to calculate an intermediate cost-to-serve to be assigned to each store.

3.6 Cost-to-serve estimation

Assuming that the arc costs c_{ij} are symmetric and that they satisfy the triangle inequalities, a natural range for the cost-to-serve of store i is $[0, 2c_{i0}]$, where the upper bound is the cost of a direct delivery to store i.

When we set $F_{ti} = 0$ for all stores, we obtain an algorithm that we call DE_0: in this case, the SDP relations tend to yield positive delivery quantities to all stores in all periods, even though such a high delivery frequency may not be necessary. Indeed, it provides very fresh products but ignores, and therefore implicitly increases, the routing costs.

When the delivery quantity to store i is close to the vehicle capacity Q, so that no other store can be served on the same route, $F_{ti} = 2c_{i0}$ is the correct delivery cost. In this case, each store could be dealt with independently and resorting to IRP solution methods no longer makes sense, at least as long as split deliveries are not allowed. In other cases, i.e., when more than one store is served by each route, neither $F_{ti} = 0$ nor $F_{ti} = 2c_{i0}$ proves to be good settings. In the sequel, we introduce two methods to calculate an intermediate cost-to-serve to be assigned to each store. The first approach yields a *distance-based cost-to-serve* F_{ti}^d which focuses on the average distance between each store and its closest neighbors. The second approach produces a *route-based cost-to-serve* F_{ti}^r which allocates the total cost of a route to the stores it includes.

3.6.1 Distance-based cost-to-serve

The first approach to estimate costs-to-serve looks at the average distance between each store and its "closest neighbors". Defining J_i as a set of stores near store i, a distance-based cost-to-serve for store i is calculated by Equation (3.10):

$$F_{ti}^d = \frac{\sum_{j \in J_i} c_{ij}}{|J_i|} \tag{3.10}$$

Observe that F_{ti}^d does not depend on t. Our experimental results show that the size of the set J_i should increase with the maximum shelf life L. The reason is that for large L, stores are served less frequently. So, there is a smaller chance to serve store i and its nearest neighbors in the same period, and, by way of consequence, it is more likely that store i will be served together with some of its farther neighbors. We set $|J_i| = 2L$ in our experiments.

3.6.2 Route-based cost-to-serve

Our second approach is inspired by the work of Ozener et al. [2013]. These authors introduce several methods to allocate a cost-to-serve to each store in an IRP. Based on the same underlying concepts, we assign the whole cost of a route to the stores it includes. However, we have to estimate the cost of the routes before solving any IRP. This can be done by calculating and comparing the average routing cost plus the average deterioration cost for different frequencies of deliveries. Assuming that the ideal periodicity of delivery is λ periods, that store i is served in period t, and that the store capacity is large enough, the delivery quantity in period t to store i can be estimated by Equation (3.11):

$$\alpha_{ti} = E(D_{ti}) + \ldots + E(D_{t+\lambda-1,i}). \tag{3.11}$$

Then, the average delivery quantity in period t to store i and to its neighbors is:

$$\bar{\alpha}_{ti} = \frac{\alpha_{ti} + \sum_{j \in J_i} \alpha_{tj}}{1 + |J_i|}. \tag{3.12}$$

Given the average delivery quantity $\bar{\alpha}_{ti}$ and vehicle capacity Q, we approximate the average number of stores included in the route serving store i in period t as:

$$\bar{n}_{ti} = \frac{Q}{\bar{\alpha}_{ti}}, \tag{3.13}$$

and the cost of the route serving i as:

$$R_{ti} = 2c_{i0} + (\bar{n}_{ti} - 1)F^d_{ti}. \tag{3.14}$$

Finally, the portion of the estimated cost of the route allocated to store i is:

$$F^r_{ti} = \frac{\alpha_{ti}}{\alpha_{ti} + \sum_{j \in J_i} \alpha_{tj}} \cdot R_{ti}. \tag{3.15}$$

In our experiments, we determined that the best setting for the frequency of deliveries is $\lambda = L - 1$. We also observed that the costs F^r_{ti} computed by this approach are high as compared to the former costs F^d_{ti}.

3.7 A decomposition-integration method

In this section, we improve our estimate of the expected profit given in Equation (3.8), by taking into account the actual routing costs in period t and by refining the approximation of the routing costs in period $t+1$ (as compared to the costs-to-serve F_{ti}). The cost-to-serve estimates are still used from period $t+2$ onward. To understand how this can be achieved, note that given the state of the system at time t, say X_t, and the vector of delivery quantities denoted by Y_t, a first estimate of the total profit for periods t to T is simply obtained as:

$$\pi^1_t(X_t, Y_t) = \sum_{i=1}^{N} f_{ti}(X_{ti}, y_{ti}) = \sum_{i=1}^{N} f_{ti}((x_1, \ldots, x_{L-1})_{ti}, y_{ti}). \tag{3.16}$$

Now, let $R(y_1, \ldots, y_N)$ represent the optimal routing cost for the VRP with delivery quantities $(y_1, \ldots, y_N)$. Then, Equation (3.16) can be improved if we replace the fixed costs-to-serve by the actual VRP routing cost in period t. This correction leads to the (presumably more accurate) estimate:

$$\pi^2_t(X_t, Y_t) = \sum_{i=1}^{N} f_{ti}(X_{ti}, y_{ti}) + \sum_{i=1}^{N} F_{ti} \cdot \mathbb{1}(y_{ti} > 0) - R(y_{t1}, \ldots, y_{tN}) \tag{3.17}$$

In order to apply a similar correction to the routing costs for period $t+1$, let us denote by $y^+_{t+1,1}, \ldots, y^+_{t+1,N}$ the optimal delivery quantities in period $t+1$. Note that these quantities depend in a complex way on (X_t, Y_t) and are actually random variables, since they also depend on the realization of the demands $D_{t1}, \ldots, D_{tN}$ in period t. With these notations, another estimate of the total expected profit can be derived from Equation (3.17), as follows:

$$\begin{aligned}
\pi^3_t(X_t, Y_t) &= \sum_{i=1}^{N} f_{ti}(X_{ti}, y_{ti}) + \sum_{i=1}^{N} F_{ti} \cdot \mathbb{1}(y_{ti} > 0) - R(y_{t1}, \ldots, y_{tN}) \\
&+ \sum_{i=1}^{N} F_{t+1,i} \cdot Pr(y^+_{t+1,i} > 0 \,|\, Y_t) \\
&- \sum_{(y_1,\ldots,y_N)} Pr((y^+_{t+1,1}, \ldots, y^+_{t+1,N}) = (y_1, \ldots, y_N)|Y_t) \times R(y_1, \ldots, y_N).
\end{aligned} \tag{3.18}$$

In this expression, the fourth term corrects the expected value of the cost-to-serve in period $t+1$, and the last term represents the expected value of the routing cost in period $t+1$, given the delivery decisions Y_t. We estimate the random variable $y^+_{t+1,i}$ by its expected value. Based on Equation (3.7), this can be estimated as follows (compare with Equation (3.8)):

$$
\begin{aligned}
&E(y^+_{t+1,i}\,|\,y_{ti}) = Pr(D_{ti} > I_{ti} + y_{ti})\, y^*_{t+1,i}(0,\ldots,0,0) \\
&+ \sum_{d=0}^{I_{ti}+y_{ti}} Pr(D_{ti} = d)\, y^*_{t+1,i}(X_{t+1,i})
\end{aligned}
\tag{3.19}
$$

where $X_{t+1,i}$ is defined by Equation (3.3).

Replacing the random quantities $y^+_{t+1,i}$ by $\lfloor E(y^+_{t+1,i}\,|\,y_{ti})\rfloor$ turns (3.18) into a deterministic problem where the delivery cost in period $t+1$ can be approximated by solving a single VRP. This approach has the drawback, however, of yielding strictly positive values $\lfloor E(y^+_{t+1,i}\,|\,y_{ti})\rfloor$ for almost all stores i, which is unlikely to happen for the optimal delivery quantities $y^+_{t+1,i}$ because this would result in high routing costs. Therefore, we further modify our approximation by considering delivery quantities $(\tilde{y}_{t+1,i}\,|\,y_{ti})$ defined by Equation (3.20) hereunder, where ε_i is a user-parameter whose value depends on the magnitude of the demand ($\varepsilon_i = \frac{1}{2}E(D_{it})$ proved suitable in our numerical experiments):

$$
(\tilde{y}_{t+1,i}|y_{ti}) = \begin{cases} \lfloor E(y^+_{t+1,i}|y_{ti})\rfloor & \text{if } \lfloor E(y^+_{t+1,i}|y_{ti})\rfloor > \varepsilon_i, \\ 0 & \text{otherwise.} \end{cases}
\tag{3.20}
$$

The optimal expected total profit is then approximated by solving the optimization Problem (3.21).

$$
\begin{aligned}
&\max_{(y_{t1},\ldots,y_{tN})} \tilde{\pi}(y_{t1},\ldots,y_{tN}) = \sum_{i=1}^{N} f_{ti}(X_{ti}, y_{ti}) \\
&+ \sum_{i=1}^{N} (F_{ti}\cdot \mathbb{1}(y_{ti} > 0) + F_{t+1,i}\cdot \mathbb{1}((\tilde{y}_{t+1,i}|y_{ti}) > 0)) \\
&- R(y_{t1},\ldots,y_{tN}) - R((\tilde{y}_{t+1,1}|y_{t1}),\ldots,(\tilde{y}_{t+1,N}|y_{tN})) \\
&\text{subject to} \quad y^{(1)}_{ti} \le y_{ti} \le C_i - I_{ti}, \;\; i = 1,\ldots,N.
\end{aligned}
\tag{3.21}
$$

This formulation takes into account the routing costs in period t, the approximated expected routing costs in period $t+1$, and costs-to-serve for the following periods. Note that by choosing a big value for ε_i in Equation (3.20), Problem (3.21) is simplified to Equation (3.17) to be maximized over $(y_{t1},\ldots,y_{tN})$, i.e., a problem considering the routing costs in period t and cost-to-serve estimates in periods $t+1$ and onward.

The decomposition-integration algorithm (DI)
Begin

Step 0. Set a cost-to-serve, F_{ti}, for each store i and each period t based on one of the algorithms described in Section (3.6). Set $t = 1$.

Step 1. Solve Problem (3.21) to obtain the delivery quantities y_{ti} and the corresponding routes in period t.

Step 2. Serve the stores with the delivery quantities y_{ti} through the routes obtained in Step 1.

Step 3. For each store i, observe the actual demand in period t, say d_{ti}. Calculate the state of the system in period $t+1$, i.e., $X_{t+1,i}$ by Relations (3.3). Set $t = t+1$ and go to Step 1.

End

In the next section, we propose a matheuristic algorithm to solve Problem (3.21), as required by Step 1 of Algorithm *DI*.

3.8 A matheuristic algorithm for Problem (3.21)

Optimizing the objective function $\tilde{\pi}$ in Problem (3.21) is a daunting task and therefore, we use a matheuristic local search algorithm to maximize it approximately. Starting from the initial solution $Y_t = (y^*_{t1}, \ldots, y^*_{tN})$ proposed by Equation (3.7), we generate a new feasible solution $Y_t^{'} = (y_{t1}^{'}, \cdots, y_{tN}^{'})$ as explained below, and we explore whether $\tilde{\pi}(Y_t^{'})$ is larger than $\tilde{\pi}(Y_t)$. If so, we move to the new solution; otherwise, we generate another solution. The local improvement algorithm stops when a pre-defined number of consecutively generated new solutions are rejected due to either lack of improvement or infeasibility. Calculating $\widetilde{\pi}$ in Problem (3.21) for any new solution involves solving two VRPs. In order to avoid these expensive computations, each new solution is not generated randomly but in a systematic way which allows us to recompute $\tilde{\pi}(Y_t^{'})$ incrementally, by difference with $\tilde{\pi}(Y_t)$.

When moving from the current solution (the current set of delivery quantities and their optimal routes) to a new solution, the difference in the approximated expected total profit is calculated by Equation (3.22):

$$\begin{aligned}
&\Delta = \widetilde{\pi}(y_{t1}^{'}, y_{t2}^{'}, \ldots, y_{tN}^{'}) - \widetilde{\pi}(y_{t1}, y_{t2}, \ldots, y_{tN}) = \\
&\sum_{i=1}^{N} (f_{ti}((x_1, x_2, \ldots, x_{L-1})_{ti}, y_{ti}^{'}) - f_{ti}((x_1, x_2, \ldots, x_{L-1})_{ti}, y_{ti})) \\
&- (R(y_{t1}^{'}, \ldots, y_{tN}^{'}) - R(y_{t1}, \ldots, y_{tN})) \\
&- (R((\tilde{y}_{t+1,1}^{'}|y_{t1}^{'}), \ldots, (\tilde{y}_{t+1,N}^{'}|y_{tN}^{'})) - R((\tilde{y}_{t+1,1}|y_{t1}), \ldots, (\tilde{y}_{t+1,N}|y_{tN}))) \\
&+ \sum_{i=1}^{N} F_{ti} \cdot (\mathbb{1}(y_{ti}^{'} > 0) - \mathbb{1}(y_{ti} > 0)) \\
&+ \sum_{i=1}^{N} F_{ti} \cdot (\mathbb{1}((\tilde{y}_{t+1,i}^{'}|y_{ti}^{'}) > 0) - \mathbb{1}((\tilde{y}_{t+1,i}|y_{ti}) > 0)).
\end{aligned} \tag{3.22}$$

Let us assume that in every move from the current solution to a new solution, we change the delivery quantities in such a way that the routes, and so the routing costs, in period $t+1$ do not change. Moreover, let us indicate the decrease in the routing costs in period t by δ:

$$\delta = R(y_{t1}, \ldots, y_{tN}) - R(y_{t1}^{'}, \ldots, y_{tN}^{'}). \tag{3.23}$$

Then, one can rewrite Equation (3.22) as follows:

$$\begin{aligned}
\Delta = &\sum_{i=1}^{N} (f_{ti}((x_1, x_2, \ldots, x_{L-1})_{ti}, y_{ti}^{'}) - f_{ti}((x_1, x_2, \ldots, x_{L-1})_{ti}, y_{ti})) \\
&+ \delta + \sum_{i=1}^{N} F_{ti} \cdot (\mathbb{1}(y_{ti}^{'} > 0) - \mathbb{1}(y_{ti} > 0)).
\end{aligned} \tag{3.24}$$

In the proposed matheuristic, a new solution is generated in such a way that the routing cost in period t decreases while the expected routing cost in period $t+1$ does not change. We explain the main idea here. Assume that in period t, a store j is ejected from its current route and is inserted into another route, say route r^*, without modification of the delivery quantities Y_t. Denote by r' the route in period t derived from route r^* after inserting store j into it. If route r' is feasible for the delivery quantities Y_t and if the routing cost in period t decreases as a result of this ejection-insertion step, then, clearly, the new VRP solution is preferred to the previous one. Generally speaking, however, since the VRP routes have been optimally selected for the delivery quantities Y_t, route r' will be infeasible either with respect to its maximum allowed length or with respect to the capacity of the vehicle. In the first case, we simply reject the new solution. In the second case, we try to determine whether the delivery quantities Y_t can be adapted (presumably, decreased) in such a way that r' becomes feasible. However, modifying Y_t also induces an effect on period $t+1$ (more precisely, on the quantities $(\tilde{Y}_{t+1}|Y_t)$ which are likely to increase). In order to keep some control over this effect, therefore, we restrict ourselves to certain modifications of Y_t which do not affect the feasibility of the current routes in period t and period $t+1$.

To describe our local search strategy, let us introduce additional notations. For an arbitrary set of routes R, we denote by $N(R)$ the set of stores contained in some route of R (excluding the depot); when R contains a single route, say, $R=\{r\}$, we simply write $N(r)$ instead of $N(R)$. Then, we define:

- R_t, R_{t+1} are the sets of routes in period t and $t+1$, respectively, after the ejection-insertion step has been performed on store j;
- $D=N(r')$ is the set of stores visited on route r'; we allow their delivery quantities to decrease in period t so as to restore feasibility of route r' (D is for "decrease");
- $\overline{R}_{t+1}=\{r\in R_{t+1}\,|\,D\cap N(r)\neq\emptyset\}$ is the set of routes in period $t+1$ that contain at least one store in D; these are the routes in period $t+1$ which may be affected when we decrease a delivery quantity to a store in D in period t; we need to make sure that these routes remain feasible, and this can be achieved by decreasing the expected delivery quantities to some of the corresponding stores in period $t+1$; or indirectly, by increasing the deliveries to these stores in period t; we model this through the introduction of the sets $\overline{R}_t$ and I;
- $\overline{R}_t=\{r\in R_t\,|\,N(\overline{R}_{t+1})\cap N(r)\neq\emptyset\}$ is the set of routes in period t that contain at least one store in $N(\overline{R}_{t+1})$; the routes in $\overline{R}_t$ are considered as being potentially affected in period t;
- $I=(N(\overline{R}_{t+1})\cap N(\overline{R}_t))\setminus D$ is the set of stores (excluding stores in D) in the affected routes in both periods t and $t+1$; we allow their delivery quantities to increase in period t so as to maintain the feasibility of the routes in $\overline{R}_{t+1}$ (I is for "increase").

Moreover, define the following binary decision variables:

- for each store $i\in D$, $v_{ih}=1$ if the delivery quantity to store i decreases by h units; else, $v_{ih}=0$;
- for each store $i\in I$, $v_{ih}=1$ if the delivery quantity to store i increases by h units; else, $v_{ih}=0$.

Assume that $\underline{m}_i$ (resp., $\overline{m}_i$) is an upper bound on the largest possible decrease (resp., increase) of the delivery quantity y_{ti} in period t. We will later explain how such bounds can be computed. Then, an IP model to determine new delivery quantities in period t is set as follows:

$$\max \sum_{i\in D}\sum_{h=0}^{\underline{m}_i} f_{ti}(X_{ti}, y_{ti}-h)\cdot v_{ih} + \sum_{i\in I}\sum_{h=0}^{\overline{m}_i} f_{ti}(X_{ti}, y_{ti}+h)\cdot v_{ih} \tag{3.25}$$

subject to

$$\sum_{h=0}^{\underline{m}_i} v_{ih} = 1 \qquad \forall i \in D \tag{3.26}$$

$$\sum_{h=0}^{\overline{m}_i} v_{ih} = 1 \qquad \forall i \in I \tag{3.27}$$

$$\sum_{i\in D}\sum_{h=0}^{\underline{m}_i} (y_{ti}-h)\cdot v_{ih} \leq Q \tag{3.28}$$

$$\sum_{i\in N(r)\cap I}\sum_{h=0}^{\overline{m}_i} (y_{ti}+h)\cdot v_{ih} + \sum_{i\in N(r)\backslash I} y_{ti} \leq Q \qquad \forall r \in \bar{R}_t \backslash r' \tag{3.29}$$

$$\begin{aligned}&\sum_{i\in N(r)\cap D}\sum_{h=0}^{\underline{m}_i} (\tilde{y}_{t+1,i}|y_{ti}-h)\cdot v_{ih} + \sum_{i\in N(r)\cap I}\sum_{h=0}^{\overline{m}_i} (\tilde{y}_{t+1,i}|y_{ti}+h)\cdot v_{ih}\\ &+ \sum_{i\in N(r)\backslash(D\cup I)} (\tilde{y}_{t+1,i}|y_{ti}) \leq Q \qquad \forall r \in \bar{R}_{t+1}\end{aligned} \tag{3.30}$$

$$v_{ih} \in \{0,1\} \qquad \forall i \in D, h \in [0, \underline{m}_i] \text{ and } \forall i \in I, h \in [0, \overline{m}_i]. \tag{3.31}$$

The objective function (3.25) maximizes the total expected profit obtained by the new delivery quantities to the stores in sets D and I, i.e., the stores whose delivery quantities may change. Constraints (3.26) and (3.27) along with Constraints (3.31) imply that exactly one of the decision variables v_{ih} takes value 1 for each store $i \in D \cup I$. Constraint (3.28) indicates that the new delivery quantities to the stores in the expanded route r' must respect the vehicle capacity. Constraints (3.29)–(3.30) guarantee that for every affected route in period t or $t+1$, the sum of the new delivery quantities does not exceed the vehicle capacity.

If the IP has a feasible solution, the new delivery quantities to the stores in D and I are calculated by using Equations (3.32) and (3.33), respectively. Delivery quantities to other stores do not change.

$$y'_{ti} = \sum_{h=0}^{\underline{m}_i} (y_{ti}-h)\cdot v_{ih} \quad \forall i \in D \tag{3.32}$$

$$y'_{ti} = \sum_{h=0}^{\overline{m}_i} (y_{ti}+h)\cdot v_{ih} \quad \forall i \in I. \tag{3.33}$$

Note that only the routes belonging to either $\bar{R}_t$ or $\bar{R}_{t+1}$ appear in the IP formulation. Moreover, in order to decrease the current excess load on route r', we only consider in (3.25)–(3.31) a subset of promising stores (those in $D \cup I$) for which the current delivery quantities can either increase or decrease. Thus, we cannot claim that the optimal solution of Problem (3.25)–(3.31) provides the optimal adjustment of delivery quantities to restore the capacity constraint in route r'. In particular, Problem (3.25)–(3.31) may be infeasible, while there actually exists an adjustment of delivery quantities such that the capacity of route r' is not exceeded and all other routes in periods t and $t+1$ remain feasible.

Thus, summing up, our local search approach to Problem (3.21) acts as a "large neighborhood search" framework, which explores the neighborhood of the current solution by solving the IP subproblem (3.25)–(3.31). The solution of (3.25)–(3.31) hopefully yields new delivery quantities which increase the expected total profit. The following algorithm must be embedded in Step 1 of algorithm *DI*.

The Matheuristic algorithm
Begin

Step 0. Initial solution: Solve two independent VRPs for periods t and $t+1$ where the delivery quantities are respectively $y_{ti} = y^*_{ti}$ and $(\tilde{y}_{t+1,i}|y_{ti})$ calculated by Equations (3.7) and (3.20).

Step 1. Termination: If Steps 2-5 have been repeated for a predetermined number of iterations, then stop.

Step 2. Ejection-insertion: Choose two random stores j and j' which are served in period t but are not included in the same route. Assume that j is ejected from its current route and is inserted immediately before or after j', whichever leads to a lower cost for the expanded route r'. If the expanded route r' is infeasible in terms of the route length, go to Step 1.

Step 3. Saving: Calculate δ as the decrease in the routing costs in period t resulting from the ejection-insertion in Step 1. If $\delta \leq 0$, go to Step 1.

Step 4. New deliveries: If the sum of the current delivery quantities on r' does not exceed Q, go to Step 5; otherwise, solve Problem (3.25)–(3.31). If the problem does not have a feasible solution go to Step 1; otherwise, calculate the new delivery quantities by Equations (3.32)–(3.33).

Step 5. Move: Use Equation (3.24) to calculate Δ, i.e., the difference between the expected total profit for the new solution and the current solution. If $\Delta > 0$ move to the new solution. Go to Step 1.

End

The Matheuristic algorithm proposed to solve Problem (3.21) relies on decreasing the routing costs in period t while keeping the expected routes in period $t+1$ unchanged. We have tested the reverse as well, i.e., modifying the delivery quantities in period t so that the routes in period t do not change while the expected routing costs in period $t+1$ decrease. Our results show that this strategy does not perform well. This may be due to the fact that the second strategy tries to decrease the expected costs of routes which may not be realized at all in period $t+1$. In contrast, the former strategy reaps an immediate benefit by decreasing the routing costs in the current period.

Maximum decrease and increase in delivery quantities:
Following the notations, here we determine the maximum decrease (resp., increase) in delivery quantity to the stores in D (resp., I), i.e., we determine $\underline{m}_i$ for $i \in D$ (resp., $\overline{m}_i$

for $i \in I$). This helps us to restrict the number of decision variables in the IP formulation. Let us define q_{tr} as the current load on route r in period t, and $q_t(i)$ as the current load on the route which includes store i in period t.

The delivery quantity to store $i \in D$ can decrease as long as it respects TSL, i.e., $\underline{m}_i \leq y_{ti} - y_{ti}^{(1)}$, where $y_{ti}^{(1)}$ is, as before, the smallest integer delivery quantity satisfying Inequality (3.4). Moreover, there is no need to decrease the delivery quantity to store $i \in D$ by more than the excess load on route r', i.e., $\underline{m}_i \leq q_{tr'} - Q$. The decrease must not cause vehicle load violation in any of the routes in $\bar{R}_{t+1}$. In order to analyze the latter constraint, consider two cases. In the first case, store $i \in D$ is not included in any route in $\bar{R}_{t+1}$, i.e., $i \in D \backslash N(\bar{R}_{t+1})$. For such a store, the delivery quantity in period t, y_{ti}, can decrease as long as the expected delivery quantity in period $t+1$, $(\tilde{y}_{t+1,i}|y_{ti})$, remains zero; otherwise, the routing costs in period $t+1$ would increase. This translates into $\underline{m}_i \leq \min_{0 \leq y \leq y_{ti}} \{y \mid (\tilde{y}_{t+1,i}|y) = 0\}$. Hence, for every store $i \in D \backslash N(\bar{R}_{t+1})$, the maximum decrease of the delivery quantity y_{ti} is determined as:

$$\underline{m}_i = \min\{y_{ti} - y_{ti}^{(1)},\ q_{tr'} - Q,\ y_{ti} - \min_{0 \leq y \leq y_{ti}} \{y \mid (\tilde{y}_{t+1,i}|y) = 0\}\}. \tag{3.34}$$

The second case considers stores $i \in D$ which are also served in period $t+1$, i.e., stores $i \in D \cap N(\bar{R}_{t+1})$. A similar reasoning about the necessity of respecting TSL and the uselessness of decreasing a delivery quantity more than the excess load on r' applies for these stores and lead to the same constraints as in the previous case. Constraints (3.30) in the IP formulation guarantee that a decrease of the delivery quantity to $i \in D \cap N(\bar{R}_{t+1})$ does not cause any vehicle capacity violation in period $t+1$. Therefore, for all stores $i \in D \cap N(\bar{R}_{t+1})$, the maximum decrease of delivery quantity is simply determined as:

$$\underline{m}_i = \min\{y_{ti} - y_{ti}^{(1)},\ q_{tr'} - Q\}. \tag{3.35}$$

A maximum increase of delivery quantity, say $\overline{m}_i$, for any store $i \in I$ can also be determined. On one hand, the increase cannot be so high as to exceed the vehicle capacity, i.e., $\overline{m}_i \leq Q - q_t(i)$ must hold. On the other hand, store capacities must be respected, i.e., $\overline{m}_i \leq C_i - I_{ti} - y_{ti}$. As a result, the maximum increase of delivery quantity to any store $i \in I$ is determined as

$$\overline{m}_i = \min\{Q - q_t(i), C_i - I_{ti} - y_{ti}\}. \tag{3.36}$$

3.9 Full information

When assessing the performance of the above algorithms, it is interesting to consider the *value of full information*, that is, the additional expected profit that could be reaped if full information about the actual demands was available to the decision-maker. With such information in hand, the PSIRP simplifies to a deterministic PIRP. In this section, we develop a simple heuristic, to be called FI, for the resulting PIRP. From the inventory control perspective, FI should deliver in such a way that no waste is incurred and no lost sales occur. To this end, there is no difference between delivering the demand of the current period only, or the demands of two periods ahead, or the demands of λ periods ahead as long as $\lambda \leq L$, since no inventory holding cost is charged. From the routing point of view, however, it can be beneficial to serve all stores in the same periods and with larger delivery quantities. In this sense, when demands are deterministic, serving all stores every $\lambda = L$ periods sounds like an effective strategy. However, here is a caveat: $\lambda < L$ might be a better choice than $\lambda = L$ because the average filling rate of vehicles could be higher. Therefore, while we expected $\lambda = L$ to be the best choice, we have tested other λ values too. Based on our setting for Q and $E(D_{ti})$, $\lambda = 2$ (resp., 3, 3) numerically proved to be the best choice for $L = 2$ (resp., 3, 4). In the algorithm FI, we assume that the stores' capacities are large.

***The full information algorithm** (FI)*
Begin

Step 0. Set $t = 1$ and λ.

Step 1. For each store i, set $y_{ti} = d_{ti} + \ldots + d_{t+\lambda-1,i} - I_{ti}$, where d_{ti} is the deterministic demand in period t in store i.

Step 2. Solve a VRP for period t by considering delivery quantities y_{ti}, and serve the stores with these delivery quantities through the optimal VRP routes.

Step 3. For each store i, set $y_{t+1,i} = \ldots = y_{t+\lambda-1,i} = 0$. Set $t = t + \lambda$ and go to Step 1.

End

3.10 Computational study

All algorithms are coded in Java and the instances are run on an Intel Core i7 processor with 1.8GHz CPU and 8GB RAM. No time limit is imposed to any of the algorithms, but each algorithm is used to determine delivery quantities and delivery routes for a finite planning horizon.

In order to solve the VRP models that arise as subproblems in all algorithms, we use a fast but effective heuristic. The heuristic first solves the LP relaxation of a route-based formulation of the VRP by column generation [Righini and Salani, 2006]. Then, the restricted master problem obtained at the end of the column generation process is solved to optimality as an integer programming problem by calling ILOG CPLEX 12.4. Testing this heuristic on the original random instances created by Solomon [1987] showed an average optimality gap of 0.6% with respect to the exact optimal values. CPLEX is also used to solve the integer programming problems (3.25)-(3.31) described in Section 3.8.

3.10.1 Instances

For the computational experiments, the first $N = 40$ stores in the R-series random instances created by Solomon [1987] are considered with some modifications. Each route length remains limited to 230 time units, but we do not impose any time window for the stores. The vehicle capacity in the homogeneous fleet is $Q = 120$. Demands are randomly generated during a planning horizon of $T = 30$ periods.

We assume that demands from end customers to the stores, D_{ti}, are i.i.d. random variables following a binomial distribution with parameters 200 and 0.1, i.e., $D_{ti} \sim Bin(200, 0.1)$. So, the average demand is $E(D_{ti}) = 20$ for each period and for each store. Three deterministic shelf lives are analyzed, namely, $L \in \{2, 3, 4\}$.

As the original instances suggest, the fixed cost of using each vehicle is zero, and Euclidean distances represent the cost c_{ij} of traveling from store i to store j. The acquisition price and selling price per unit are respectively $a = 6$ and $p = 10$. A target service level of $TSL = 90\%$ is to be respected in every period and every store in all solution methods, except EV (note that EV does not access the probability distribution of demand). A capacity of $C_i = 120$ is considered for every store. However, our computational results reveal that when algorithms UL_λ, DE, and DI are applied, the proposed delivery quantities by the algorithms plus the inventory level is always strictly less than 40 (resp., 60, 80) for $L = 2$ (resp., 3, 4). This implies that for these algorithms the latter store capacities are large enough. When algorithm FI is applied, store capacity 80 (resp., 100, 120) is large enough for $L = 2$ (resp., 3, 4). We will later analyze the impact of limited store capacity on profit, freshness, and actual service level. In the next subsections, we discuss some of the performance measures that we have collected. Following our settings, Equations (3.10) and (3.15) suggest $F_d = \{11, 13, 15\}$ and $F_r = \{18, 25, 31\}$ for $L = \{2, 3, 4\}$.

3.10.2 Simulation

In order to evaluate the performance of different solution methods, we use random scenarios to simulate the sequence of decisions made by each method over a planning horizon of $T = 30$ periods. We generate a set of 30 scenarios, where each scenario consists of initial inventory as well as demands of the stores over the planning horizon. We use the same scenarios for all solution methods and for all shelf lives. The initial inventory of each store is a uniform random number in the interval $[0, 30]$ (resp., $[0, 50]$, $[0, 70]$) for $L = 2$ (resp., 3, 4), and is considered to have shelf life $L - 1$.

For each solution method, the expected profit is estimated by averaging the total profit over the 30 random scenarios. Since each algorithm faces the same sequence of demands in each scenario, we can meaningfully compare the profits associated with the decisions made by each algorithm with regard to delivery quantities and delivery routes. Besides, we also collect other useful information such as average actual service level and freshness as additional criteria to measure the performance of each method.

3.10.3 Actual service level

For each run of the simulation, we calculate the average actual service level in two ways, based on the number of stock-outs (ξ_s) and fill rate (ξ_f). Recall that, according to Equation (3.1), I_{ti} indicates the inventory level at the beginning of period t in store i, i.e., the inventory level before delivery. The quantity $\mathbb{1}(d_{ti} \leq I_{ti} + y_{ti})$ is 1 if no stock-out happens in period t in store i, and 0 otherwise. Hence, in Equation (3.37) hereunder, ξ_s is the proportion of observations where no stock-outs occurred, over all stores and all periods. This metric is consistent with our initial definition of TSL in Equation (3.4).

$$\xi_s = \frac{\sum_t \sum_i \mathbb{1}(d_{ti} \leq I_{ti} + y_{ti})}{TN}. \tag{3.37}$$

Our second definition of service level considers the fill rate of demands. In this case, $\min\{d_{ti}, I_{ti} + y_{ti}\}$ shows the demand satisfied in period t in store i. Thus, Equation (3.38) calculates the average fill rate of demand in all stores over the planning horizon.

$$\xi_f = \frac{\sum_t \sum_i \min\{d_{ti}, I_{ti} + y_{ti}\}}{\sum_t \sum_i d_{ti}}. \tag{3.38}$$

3.10.4 Actual freshness

For each run of the simulation, the actual freshness of products is calculated in two ways. First, the average actual freshness on shelf ϕ_s is calculated by Equation (3.39):

$$\phi_s = \frac{\sum_t \sum_i (1x_1 + 2x_2 + \cdots + (L-1)x_{L-1})_{ti} + Ly_{ti}}{\sum_t \sum_i (I_{ti} + y_{ti})}. \tag{3.39}$$

Next, Equation (3.40) is used to calculate ϕ_c, the average actual freshness from a customer's perspective. In this definition, $(s_k)_{ti}$ is the number of units with remaining shelf life k sold in period t in store i.

$$\phi_c = \frac{\sum_t \sum_i (1s_1 + 2s_2 + \cdots + (L-1)s_{L-1} + Ls_L)_{ti}}{\sum_t \sum_i (s_1 + s_2 + \cdots + s_L)_{ti}}. \tag{3.40}$$

In Equation (3.40), values of s_k are determined by the following recursive relations:

$$\begin{array}{ll} u_1 = d & s_1 = \min(u_1, x_1) \\ u_2 = (u_1 - x_1)^+ & s_2 = \min(u_2, x_2) \\ u_3 = (u_2 - x_2)^+ & s_3 = \min(u_3, x_3) \\ \vdots & \vdots \\ u_L = (u_{L-1} - x_{L-1})^+ & s_L = \min(u_L, y) \end{array} \tag{3.41}$$

3.10.5 Verifying route schedule estimations in period $t+1$

In the decomposition-integration method *DI*, we use Equation (3.20) to approximate the expected deliveries in period $t+1$. The main purpose of this approximation is to estimate the routing costs in period $t+1$; see Equation (3.21). Therefore, the accuracy of the approximation can be evaluated for each scenario by measuring the similarity between the set of routes forecasted when using Equation (3.20), denoted here by $E(R_{t+1})$, and the set of routes actually used in period $t+1$, namely, R_{t+1}. We define the degree of similarity between these sets by Equation (3.42):

$$Similarity = \frac{\sum_{(i,j)\in(R_{t+1}\cap E(R_{t+1}))} c_{ij}}{\sum_{(i,j)\in(R_{t+1}\cup E(R_{t+1}))} c_{ij}}. \tag{3.42}$$

3.10.6 Results

For each scenario, each solution method is applied over a planning horizon of $T = 30$ periods. Tables 3.3 and 3.4 summarizes the results. The first column denotes the maximum shelf live $L \in \{2,3,4\}$. The second column indicates the solution methods applied to determine delivery quantities and routes for each scenario, namely: the expected value method (EV), deliver-up-to-level with daily deliveries (UL_1), deliver-up-to-level with large delivery quantities to satisfy TSL for $\lambda = L-1$ periods (UL_{L-1}), decomposition without costs-to-serve (DE_0), decomposition with distance-based costs-to-serve (DE_d), decomposition with route-based costs-to-serve (DE_r), decomposition-integration without costs-to-serve (DI_0), decomposition-integration with distance-based costs-to-serve (DI_d), decomposition-integration with route-based costs-to-serve (DI_r), and the full information method (FI). Column 3 displays the average computation times over 30 scenarios for each instance. When $L = 2$, most of the computation time is spent in solving the VRPs, in that all $N = 40$ stores are served in every period when applying UL_λ, *DE*, or *DI*. When $L = 4$, however, most of the computation time is devoted to solving the expensive SDP relations (3.8). In the latter case, solving the VRPs takes almost no time because the average number of stores served in each period is around 15. The next columns report, respectively, average values over 30 scenarios of the profit, revenue, acquisition cost, routing cost, waste cost, average number of vehicles per period, average number of stores per route, average time between two consecutive visits to stores, freshness on shelf, freshness from customers' perspective, service level based on the number of stock-outs, and service level based on the filling rate.

In order to increase the readability of the table, the values in Columns 4–8 are normalized with respect to the profit obtained by EV for each shelf life. For UL_{λ}, we obtained the highest profit by setting $\lambda = L-1$. For FI, we obtained the highest profit by setting $\lambda = 2$ (resp., 3, 3) for $L = 2$ (resp., 3, 4).

3.11 Discussion

In this section, we analyze and discuss the results of the computational study. We demonstrate that the differences between the profits generated by different solution methods are statistically and economically significant. We also draw some additional managerial insights.

3.11.1 Comparing the solution methods

Before we focus on the profit criterion, let us briefly discuss the other trends that emerge from Tables 3.3 and 3.4. Regarding freshness, method UL_1 provides the freshest products, while the other solution methods strive to reap a higher profit. All methods lead to extremely high service levels, especially when measured by the fill rate of demand ξ_f. These high service levels are obtained even for EV, especially when L is large. As indicated by ξ_s, the proportion of stock-outs is higher with EV and UL_1 than with the other methods, but is still reasonably low. Finally, all solution methods but EV yield extremely low waste costs.

Table 3.3 reveals that the computation time of *DI* is always more than that of *DE*, as the former algorithm builds upon the solution of the latter one. Interestingly, the computation time of *DE* and *DI* with positive cost-to-serve values for $L = 3$ are less than that for $L = 2$ and 4. The reason is that for a larger value of L the SDP computations take more time, but the stores are visited less frequently. This implies that a fewer number of stores are visited in each period with larger delivery quantities, which consequently facilitates solving the VRPs. Indeed, when $L = 2$, the SDP computations are carried out very quickly, whereas solving the VRPs take a lot of time, as almost all stores are visited in every period. On the contrary, when $L = 4$, the SDP computations take most of the computation time, whereas the VRPs are solved more conveniently. In this sense, when $L = 3$, the sum of the computation time to solve the SDP relations and the VRPs is less than $L = 2$ and 4.

Let us now take a closer look at the profit. Recall that all values are normalized so that the expected profit generated by EV is 1 in all scenarios. As expected, the average profit tends to increase when we move from method EV to UL, to *DE*, and to *DI*. Figure 3.1a illustrates the average additional profit when each of the methods is applied.

Table 3.3: Comparing different solution methods

L	Method	Time(sec)	Pro.=	Rev.	$-$Acq.	$-$Rou.	Waste
	EV	4	1.000	3.598	2.331	0.267	0.149
	UL_1	1626	1.126	3.596	2.175	0.295	0.008
	UL_{L-1}	1626	1.126	3.596	2.175	0.295	0.008
	DE_0	1802	1.129	3.623	2.196	0.297	0.011
2	DE_d	1484	1.130	3.623	2.196	0.297	0.011
	DE_r	1482	1.130	3.623	2.196	0.297	0.011
	DI_0	8783	1.133	3.622	2.196	0.293	0.011
	DI_d	7003	1.133	3.621	2.196	0.293	0.011
	DI_r	7188	1.133	3.621	2.196	0.293	0.011
	FI	7	1.210	3.604	2.170	0.224	0.007
	EV	3	1.000	3.376	2.164	0.212	0.103
	UL_1	1673	1.045	3.310	1.994	0.271	0.000
	UL_{L-1}	2	1.082	3.379	2.065	0.232	0.011
	DE_0	1575	1.052	3.359	2.032	0.274	0.000
3	DE_d	49	1.082	3.382	2.073	0.227	0.016
	DE_r	50	1.078	3.384	2.079	0.227	0.020
	DI_0	7697	1.057	3.356	2.036	0.263	0.005
	DI_d	98	1.088	3.384	2.073	0.223	0.015
	DI_r	98	1.085	3.383	2.077	0.221	0.019
	FI	1	1.132	3.317	1.995	0.189	0.005
	EV	0	1.000	3.418	2.182	0.236	0.084
	UL_1	1672	1.046	3.315	1.997	0.272	0.000
	UL_{L-1}	0	1.109	3.423	2.110	0.205	0.014
	DE_0	2568	1.054	3.364	2.036	0.275	0.000
4	DE_d	1086	1.111	3.425	2.109	0.205	0.012
	DE_r	1107	1.106	3.431	2.121	0.204	0.018
	DI_0	7716	1.070	3.362	2.039	0.253	0.005
	DI_d	1365	1.111	3.424	2.104	0.210	0.007
	DI_r	1597	1.117	3.426	2.113	0.197	0.014
	FI	1	1.138	3.322	1.998	0.186	0.005

Table 3.4: Comparing different solution methods

L	Method	Veh.	Cus.	Bet.	ϕ_s	ϕ_c	ξ_s	ξ_f
	EV	8.5	3.2	1.6	1.6	1.4	91%	98%
	UL_1	7.7	5.6	1.0	1.7	1.7	93%	99%
	UL_{L-1}	7.7	5.6	1.0	1.7	1.7	93%	99%
	DE_0	7.8	5.5	1.0	1.7	1.6	97%	99%
2	DE_d	7.8	5.5	1.0	1.7	1.6	97%	99%
	DE_r	7.8	5.3	1.0	1.7	1.6	97%	99%
	DI_0	7.5	5.5	1.0	1.7	1.6	97%	99%
	DI_d	7.6	5.7	1.0	1.7	1.6	97%	99%
	DI_r	7.5	5.5	1.0	1.7	1.6	97%	99%
	FI	7.8	2.6	2.0	1.6	1.5	100%	100%
	EV	8.5	2.0	2.6	2.3	1.9	95%	99%
	UL_1	7.7	5.6	1.0	2.7	2.7	93%	99%
	UL_{L-1}	8.1	3.1	1.8	2.4	2.0	99%	99%
	DE_0	7.7	5.5	1.0	2.6	2.3	99%	99%
3	DE_d	8.2	2.8	1.9	2.3	1.9	99%	99%
	DE_r	8.2	2.0	2.0	2.3	1.9	99%	99%
	DI_0	7.5	5.5	1.0	2.6	2.3	99%	99%
	DI_d	7.8	3.0	1.9	2.4	2.0	99%	99%
	DI_r	7.8	2.0	2.0	2.3	1.9	99%	99%
	FI	8.0	1.7	3.0	2.3	2.0	100%	100%
	EV	11.7	1.0	3.6	3.0	2.5	97%	99%
	UL_1	7.7	5.6	1.0	3.7	3.7	93%	99%
	UL_{L-1}	8.2	1.0	2.8	3.1	2.5	99%	99%
	DE_0	7.7	5.5	1.0	3.6	3.3	99%	99%
4	DE_d	8.2	1.9	2.8	3.1	2.5	99%	99%
	DE_r	8.2	1.4	2.9	3.0	2.4	99%	99%
	DI_0	7.4	5.3	1.1	3.5	3.2	99%	99%
	DI_d	7.9	2.6	2.2	3.2	2.7	99%	99%
	DI_r	7.7	2.1	2.6	3.1	2.5	99%	99%
	FI	8.0	1.7	3.0	3.3	2.9	100%	100%

When the demand distribution is known, DE_0 can be applied, whereby the profit increases on average by 12.9% (resp., 5.2%, 5.4%) for $L = 2$ (resp., 3, 4). This increase can be interpreted as the value of accessing the probability distribution and of explicitly accounting for the uncertainty of demand. The additional gap filled by DE_d shows the value of considering some aspect of routing when we determine delivery quantities. On average, it amounts to an additional increase in profit of 0.1% (resp., 3.0%, 5.7%) for $L = 2$ (resp., 3, 4). Then, by using DI_d the profit increases again on average by 0.3% (resp., 0.6%, 0.0%) for $L = 2$ (resp., 3, 4): this measures the value of further integrating inventory and routing-related decisions. Finally, accessing full information and applying FI provides some 7.7% (resp., 4.4%, 2.7%) average additional profit for $L = 2$ (resp., 3, 4). This can be interpreted as the *value of full information*. Figure 3.1a also shows the profit gained by applying UL_{L-1}. Interestingly, the performance of this simple deliver-up-to-level policy is very close to the performance of the more sophisticated method DE. Figure 3.1b can be interpreted in the same way as the previous figure, when the cost-to-serve F^d is replaced by F^r.

We have also tested the performance of *DI* if the profit function in Problem (3.21) is replaced by a simpler estimate, say, (3.17). In the latter estimation, costs-to-serve $F_{t+1,i}$ are considered for period $t+1$, whereas Problem (3.21) calculates a VRP routing cost for period $t+1$ based upon the expected delivery quantities. Solving (3.17) is easier than (3.21), but our computational results show that the profit generated by (3.17) falls between the profits generated by *DE* and by *DI*.

3.11.2 Statistical tests

Algorithm *DI* takes the solution provided by *DE* as an initial solution and tries to improve it by applying a Matheuristic. The resulting average improvement of profit is apparently small, but proves statistically significant as we demonstrate next. Let us first consider the cost-to-serve F^d and the associated methods DE_d and DI_d. We test the statistical hypothesis $H_0 : P_{DI_d} \leq P_{DE_d}$, where P_{DE_d} and P_{DI_d} indicate the total profits obtained by DE_d and DI_d, respectively. The results of the t-test for paired samples are displayed in Table 3.5, where $\bar{P} = P_{DI_d} - P_{DE_d}$. Recall that for paired samples, t-statistic is calculated as:

$$t - statistic = \frac{E(P_{DI_d} - P_{DE_d})}{Std(P_{DI_d} - P_{DE_d})/\sqrt{n}}, \tag{3.43}$$

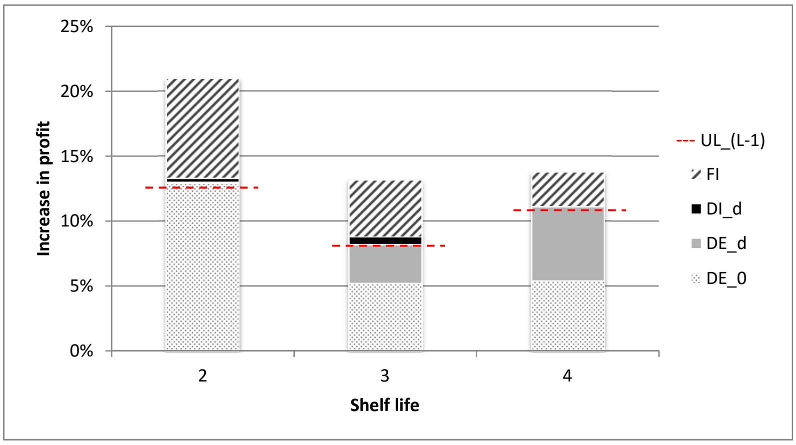
where n is the sample size, and the t-statistic has $n-1$ degrees of freedom.

The threshold for the t-statistic with 29 degrees of freedom (above which the null hypothesis is rejected with confidence level 99.99%) is $t_{0.9999;29} = 4.25$. Table 3.5 shows that H_0 is rejected in all cases but one, which shows that *DI* dominates *DE* in terms of profit.

Superiority of *DI* over *DE* is not confined to improving the profit. In particular, *DI* uses fewer vehicles than *DE*, and the difference is again statistically significant, as shown in Table 3.6, where $\bar{V} = V_{DE_d} - V_{DI_d}$. Similar conclusions apply when F^d is replaced by F^r. These results demonstrate that it makes sense to use our Matheuristic to build upon *DE*.

3.11.3 Impact of cost-to-serve values on *DE* and *DI*

So far, we have defined two ways to assign a positive cost-to-serve to a generic store, namely, F^d and F^r. In general, F^r is much larger than F^d. Therefore, we have also



(a) distance-based cost-to-serve

(b) route-based cost-to-serve

Figure 3.1: Contribution of different solution methods to increasing profit

Table 3.5: Statistical tests on profit

L	C	$E(\bar{P})$	$Std(\bar{P})$	t-statistic	Hypothesis	Result	Prof. inc.
2	30	192	53	20.0	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	0.3%
	≥40	217	62	19.2	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	0.3%
3	30	118	48	13.4	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	0.2%
	40	379	96	21.6	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	0.5%
	50	391	147	14.6	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	0.5%
	≥60	408	130	17.2	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	0.5%
4	30	118	48	13.4	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	0.2%
	40	407	85	21.6	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	0.5%
	50	352	111	17.4	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	0.4%
	60	911	204	24.4	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	1.2%
	70	233	166	7.7	$H_0 : P_{DI_d} \leq P_{DE_d}$	Reject	0.3%
	≥80	30	230	0.7	$H_0 : P_{DI_d} = P_{DE_d}$	Accept	0.0%

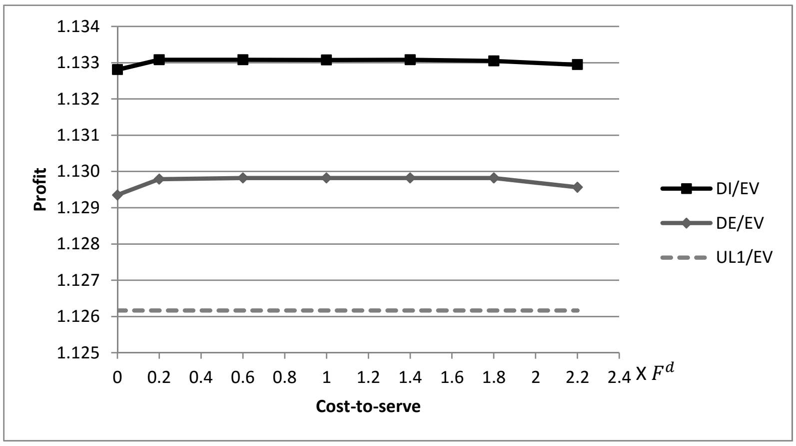
Table 3.6: Statistical tests on number of vehicles

L	C	$E(\bar{V})$	$Std(\bar{V})$	t-statistic	Hypothesis	Result	Vehi. dec.
2	30	0.22	0.09	13.4	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	2.8%
	≥40	0.21	0.08	14.4	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	2.8%
3	30	0.13	0.11	6.5	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	1.7%
	40	0.19	0.10	10.4	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	2.5%
	50	0.32	0.10	17.5	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	3.9%
	≥60	0.34	0.11	16.9	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	4.2%
4	30	0.13	0.11	6.5	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	1.7%
	40	0.22	0.10	12.0	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	2.8%
	50	0.31	0.09	18.9	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	3.8%
	60	0.91	0.12	41.5	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	10.3%
	70	0.42	0.13	17.7	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	5.1%
	≥80	0.30	0.16	10.3	$H_0 : V_{DI_d} \geq V_{DE_d}$	Reject	3.6%

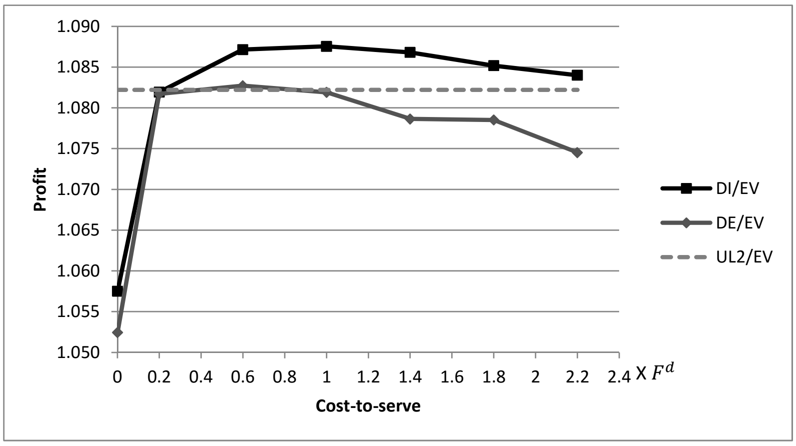
tested the sensitivity of the performance of *DE* and *DI* when other values of the cost-to-serve are considered. The results are shown in Figure 3.2. It appears that, for all three shelf lives, *DE* achieves its best performance when F^d is set as cost-to-serve. (Note that the horizontal axis is normalized so that $F^d = 1$ in all cases, and the vertical axis shows the relative profit with respect to EV.) However, when *DI* is used, the best setting of cost-to-serve is not that clear: while F^d provides the highest profit for shelf lives $L = 2$ and $L = 3$, $F^r \approx 2.2F^d$ results in the best profit for shelf life $L = 4$.

In fact, our test results show that if the average number of stores per route, say $\bar{n}$, is at least 3, then F^d works well for all values of L. On the other hand, $\bar{n} < 2$ implies that routes rarely include more than two stores. In this case, $F_i = K + 2c_{i0}$ proves a better estimation for cost-to-serve than F^d, and we can adopt a direct delivery policy for store i. This is consistent with the results in [Gallego and Simchi-Levi, 1990; Bertazzi, 2008], i.e., when the delivery quantity is a large fraction of the vehicle capacity, direct shipping is preferable in almost all routing strategies.

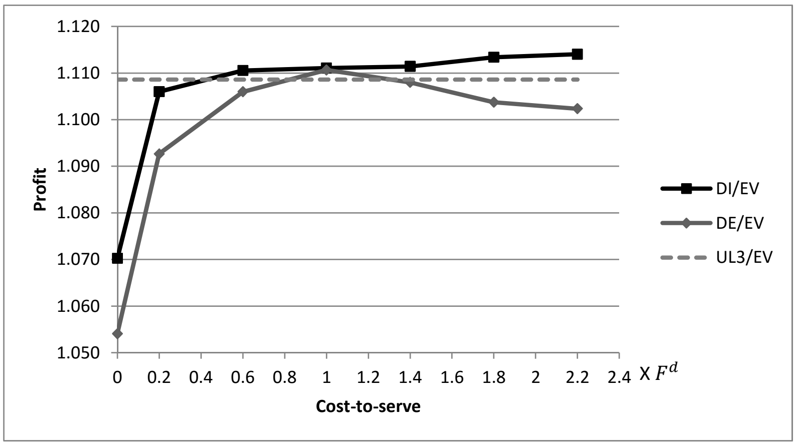
Figure 3.2 also demonstrates that, no matter what value is selected for the cost-to-serve, there is always some improvement in profit when *DI* is used.



(a) $L = 2$



(b) $L = 3$



(c) $L = 4$

Figure 3.2: Profit obtained by DE and DI with different costs-to-serve for $Q = 120$.

3.11.4 Managerial insights

In this section, we discuss the economic significance of the profit improvements, the impact of shelf life and store capacity on the profit obtained by the best solution method, and the interpretation of *DE* as an (R, s, S) policy.

Economic interpretation of profit improvements. We argue here that the improvement in profit provided by *DI* over *DE* is not only statistically, but also economically significant. In our experimental setting, profit is about 32% of total revenue, but does not account for a variety of miscellaneous costs (salaries, buildings, marketing, administration, and so forth). In fact, net profit in the retail food sector is of the order of 2% of revenue; see Euro Bank [2009]; NAICS [2012]; FMI [2014]. This means that, under our assumptions, miscellaneous costs would account for about 30% of total revenue. The corresponding breakdown of the revenue is depicted in Figure 3.3.

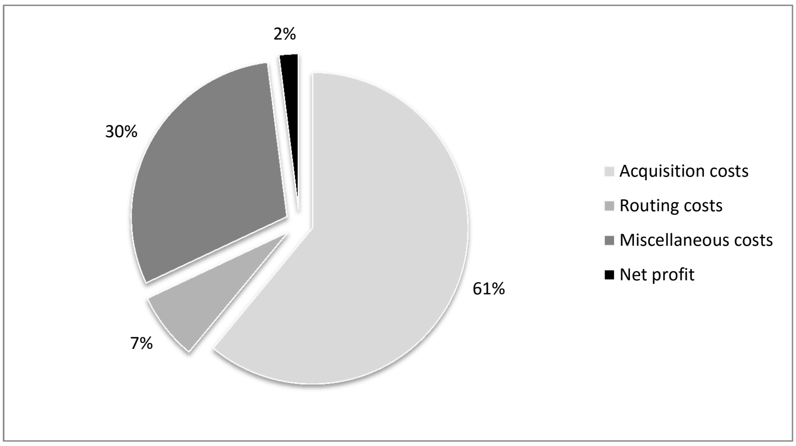


Figure 3.3: Breakdown of the revenue.

Our experimental results show an average improvement of 0.6% in profit when we exploit *DI* as compared to *DE* while setting an appropriate cost-to-serve (0.3% for $L = 2$ with F^d, 0.5% for $L = 3$ with F^d, and 1.0% for $L = 4$ with F^r). This translates into 0.19% of the revenue (= 0.6% of 32%), meaning about 10% of the net profit of a typical retail chain. This is certainly economically significant.

Impact of store capacity. Intuitively, one might expect that the smaller F_{ti}, the more frequently store i is visited. However, our experimental results show that the frequency of visits is relatively insensitive to F_{ti}, provided that F_{ti} is strictly positive. Therefore, selecting the value of the cost-to-serve cannot be regarded as a lever to adjust the frequency of visits and the freshness of products. On the other hand, the store capacity clearly has an effect on these performance indicators.

Table 3.7 shows the expected profit obtained by DI_d over $T = 30$ periods, as well as freshness and service level, when considering a limited store capacity C. We see that the service level is only slightly influenced by C. The changes in profit and freshness, however, are significant. The results suggest that providing extra store capacity beyond $(L-1)E(D) + 0.5E(D)$ does not have any major impact on profit. Observe that $(L-1)E(D)$ can be viewed as the expected required capacity between two consecutive visits (during $L-1$ periods) when the visits are maximally spread, while the quantity

$0.5E(D)$ acts as buffer inventory to respect the target service level during $L-1$ periods on average. The last column of Table 3.7 indicates the similarity values calculated by Equation (3.42). The values show that *DI* estimates reasonably well the expected routes in period $t+1$.

Table 3.7: Comparing the impact of different capacities on profit, freshness, and service level

L	C	Profit by DI_d	ϕ_s	ϕ_c	ξ_s	ξ_f	Similarity
2	30	75375	1.7	1.6	97%	99%	70%
	≥40	75394	1.7	1.6	97%	99%	71%
3	30	76216	2.7	2.5	98%	99%	69%
	40	76997	2.5	2.1	99%	99%	50%
	50	78613	2.4	2.0	99%	99%	53%
	≥60	78628	2.4	2.0	99%	99%	53%
4	30	76216	3.7	3.5	98%	99%	69%
	40	77130	3.5	3.1	99%	99%	50%
	50	79790	3.3	2.9	99%	99%	53%
	60	80108	3.2	2.8	99%	99%	72%
	70	80368	3.2	2.7	99%	99%	70%
	≥80	80368	3.2	2.7	99%	99%	70%

Impact of shelf life. Figure 3.4 shows the profits (in absolute value) obtained by DI_d for $L=2,3$ and by DI_r for $L=4$. The numbers indicate a 2.5% decrease in profit when shelf life decreases from 4 to 3. Some further 4.3% loss in profit is incurred when moving from shelf life 3 to 2. These values can be interpreted as the cost of perishability. Recall that these decreases translate into $2.5\times16=40.7\%$ and $4.3\times16=68.6\%$ decreases in the net profit, which are extremely significant.

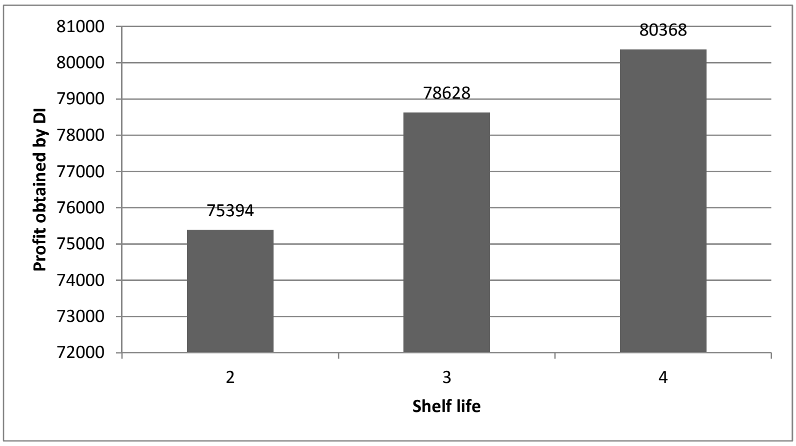


Figure 3.4: The best profit obtained by *DI* for different shelf lives.

Translating** DE **into an** (R,s,S) **policy. Interestingly, as illustrated by Figure 3.2, the simple algorithm UL_{L-1} is a strong competitor for *DE* when $L\geq3$, regardless of the value of the cost-to-serve. For some cost-to-serve values, UL_{L-1} even dominates *DE*. However, when UL_{L-1} outperforms *DE*, *DI* is able to improve the initial solution provided by DE so that the final solution is better than UL_{L-1}. In our instances, the inventory level triggering a delivery in UL_{L-1} is $s_{UL} = 1.25E(D)$ for any L (see Equation (3.4)), and the up-to-level point is $S_{UL} = 1.3E(D)$ (resp., $2.4E(D)$, $3.45E(D)$) for $L = 2$ (resp., 3, 4); see Equation (3.5). Our computational experiments reveal that the inventory level triggering a delivery in method DE is $s_{DE} = 1.25E(D)$ for any L, independently of the value of X_{ti}, the state of the system in period t in store i. On the other hand, the delivery quantities prescribed by DE do depend on X_{ti}, but mostly through the value of the total inventory level $I_{ti} = \sum_{k=1}^{L-1}(x_k)_{ti}$.

When TSL is high, the up-to-level point in DE, i.e., $S_{DE} = (I_{ti} + y^*_{ti}|X_{ti})$, is quite close to that in UL_{L-1}, i.e., $S_{UL} = (I_{ti} + y^*_{ti}|I_{ti})$, especially when L is large. The up-to-level point in DE slightly increases when setting a higher cost-to-serve. Figure 3.5 shows the normalized frequency (over all possible states X_{ti}) of values of the up-to-level point, S_{DE}, when DE_d is applied.

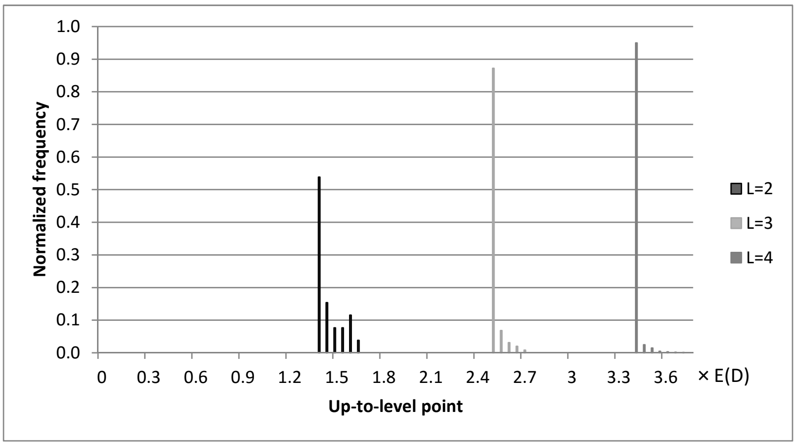


Figure 3.5: The normalized frequency of up-to-level points in DE_d for different shelf lives.

According to the figure, for all states of the system, the up-to-level point determined by DE_d is very likely to be $1.4E(D)$ (resp., $2.5E(D)$, $3.4E(D)$) for $L = 2$ (resp., 3, 4). For example, when $L = 3$ and the inventory level does not satisfy TSL in the current period, DE prescribes $S_{DE} = 2.5E(D)$ as the up-to-level point in 88% of the states, whatever the breakdown of X_{ti} is. This implies that all complex SDP Relations (3.8) can be developed once offline and be translated into a simple and easy-to-interpret (R,s,S) policy, where $R = 1$, $s = y^{(1)}_{ti}$, and $S = 1.4E(D)$ (resp., $2.5E(D)$, $3.4E(D)$) for $L = 2$ (resp., 3, 4), without any major impact on the performance of DE.

3.11.5 The main messages

All solution methods discussed in this chapter have their advantages and limitations, and each proves to perform well under different conditions. The main features of EV, UL_{L-1}, DE, and DI are summarized in Table 3.8.

Table 3.8: Features of different solution methods

Method	Feature
EV	Extremely simple Does not take the stochasticity of demands into consideration Leads to the lowest profit, service level, and freshness Leads to the highest waste cost and to the highest number of vehicles Can be regarded as a base case
UL_{L-1}	Extremely simple Its special case, UL_1, provides freshest products Useless when TSL is not defined or is low A strong competitor for DE, especially when TSL is high Applicable to multiple products
DE	$F > 0$ reflects some aspects of routing when deciding about delivery quantities F^d performs best if $\bar{n} \geq 2$; otherwise F^r performs better Performs similarly to the deliver-up-to-level policy
DI	Builds upon DE F^d delivers the best results provided that $\bar{n} \geq 3$ Decreases the routing costs in the current period, as expected Superior to DE statistically and economically, in terms of profit and number of vehicles Superiority over DE applies for a range of values of store capacity and different F_{ti} Slightly higher freshness but the same actual service level compared to DE Superior to UL_{L-1} even when UL_{L-1} dominates DE

The main messages we can draw from Table 3.8 are as follows:

- UL_{L-1} has a very good performance on various criteria, but DI significantly improves the expected profit and performs well on other criteria;
- superiority of DI to the other solution methods in terms of profit applies for a range of values of C and is statistically valid and economically significant;
- the superiority of DI applies for different estimates of the cost-to-serve values.

3.11.6 Extensions

All the solution methods but FI can be extended to account for inventory holding costs or for decaying products, i.e., products which lose their quality gradually over their shelf life. This is straighforward for methods EV and UL, but less so for DE and DI. Let us define h as the inventory holding cost per unit per period. Moreover, let us assume that the value of each unit of the product decreases by $h^{'}$ monetary units in each period. The parameter $h^{'}$ can alternatively be considered as a self-imposed penalty with the objective to increase freshness when modeling perishable products. In other words, even if the selling price is actually constant during the shelf life (perishable products), the retail chain may assume that the value of the product decreases linearly over time (decaying products) in order to enforce higher freshness. In order to incorporate these elements in DE and DI, we can add the term e_{ti} defined by Equation (3.44) hereunder to the profit function given in Equation (3.8):

$$e_{ti} = -(h+h^{'})I_{ti} + h^{'}(L-1)\sum_{d=0}^{x_{t1}-1} Pr(D_{ti}=d)(x_{t1}-d). \tag{3.44}$$

The first term in (3.44) charges the total inventory at the beginning of period t with costs h and $h^{'}$, since this inventory is carried from the previous period. The second term cancels out the charged costs $h^{'}$ during $L-1$ periods for the units which are completely deteriorated at the end of period t.

3.12 Conclusions

By considering uncertainty and combining inventory with routing decisions for perishable products, retail chains can obtain a significant increase in their net profit. We have shown how such benefit can be gained and we have quantified it. The expected value method, where only the expected demands are taken into consideration in retail chain's decisions, serves as a benchmark. We then show how the knowledge of the demand distribution can add to the profit. To this end, we first propose a simple up-to-level method which explicitly takes the target service level into account. Our numerical results show that this naive policy performs reasonably well when the target service level is high. Next, a decomposition method is applied to determine delivery quantity to each store independently. Assigning virtual costs-to-serve to stores whenever they are visited accounts for some aspects of routing in the method. This leads to a significant increase in profit. Finally, we integrate the decisions independently made by each store, and we slightly divert from the latter delivery quantities with the aim to decrease the routing costs. Though the routing costs only comprise a small portion of the total costs, we showed that the final improvement in total profit is statistically and economically significant. Our approach considers the real (expected) routes for only two periods ahead in the decomposition-integration method. This is justifiable when routing decisions cannot be made for a large number of periods and deliveries cannot be synchronized to be carried out in the same periods. This is the case when (1) demands are highly stochastic, and (2) shelf life is short or store capacity is limited for long-term deliveries. At last, we show how further profit improvement is possible when accessing full information about the future demands.

Chapter 4

A two-period vehicle routing problem with full delivery shifts

4.1 Introduction

Consider a logistics service provider (LSP) supplying units of a single product from a central warehouse to a number of geographically dispersed stores. The LSP has access to an unlimited supply of the product. Independently of other stores, each store places its orders for two successive periods, say, day $t+1$ and day $t+2$. The LSP may decide to postpone a delivery requested for period $t+1$ and to deliver instead a (possibly different) quantity in period $t+2$, while paying a financial penalty. In a similar way, advancing a delivery from period $t+2$ to period $t+1$ could be acceptable for a store, but the LSP is charged a penalty for it. In our terminology, advancements and postponements are referred to as *shifts* of deliveries. Each store specifies whether shifts are allowed and, if so, specifies the alternative delivery quantities and the associated penalties. In fact, we may assume that shifts are always allowed for all stores but that the associated penalties for some stores may be so high that they effectively deter the LSP from performing the shifts. The LSP's objective is to minimize the sum of the routing costs in two periods and of the penalties for the shifted deliveries. Compared to solving two independent VRPs, solving this two-period VRP is beneficial for both sides, i.e., the LSP and the stores. Obviously, the solution of the two-period VRP is advantageous for the LSP, as it provides more flexibility to coordinate the routing costs of two periods and consequently, to decrease their total cost. On the other hand, the stores can choose a penalty that is high enough to compensate the costs incurred by their inventory systems when deliveries are shifted.

4.1.1 Motivation 1: from the SIRP to the two-period VRP

In Section 3.8, we discussed a Matheuristic algorithm to solve Problem (3.21) based on local search. Here, we show that Problem (3.21) can be solved as a two-period VRP with delivery shifts. Assume that in the SIRP, $(y^*_{t1},\dots,y^*_{tN})$ is the vector of delivery quantities to the stores in period t calculated based on Equation (3.7). Moreover, assume that $((\tilde{y}_{t+1,1}|y^*_{t1}),\dots,(\tilde{y}_{t+1,N}|y^*_{tN}))$ is the vector of the expected delivery quantities calculated based on Equation (3.20). Problem (3.21) can be rewritten as:

$$\begin{aligned}
&\max_{(y_{t1},\dots,y_{tN})} \sum_{i=1}^{N} \{ f_{ti}((x_1,x_2,\dots,x_{L-1})_{ti}, y_{ti}) + F_{ti} \cdot \mathbb{1}(y_{ti} > 0) + F_{t+1,i} \cdot \mathbb{1}((\tilde{y}_{t+1,i}|y_{ti}) > 0) \} \\
&- \sum_{i=1}^{N} \{ f_{ti}((x_1,x_2,\dots,x_{L-1})_{ti}, y^*_{ti}) + F_{ti} \cdot \mathbb{1}(y^*_{ti} > 0) + F_{t+1,i} \cdot \mathbb{1}((\tilde{y}_{t+1,i}|y^*_{ti}) > 0) \} \\
&- R(y_{t1},\dots,y_{tN}) - R((\tilde{y}_{t+1,1}|y_{t1}),\dots,(\tilde{y}_{t+1,N}|y_{tN})) \\
&\text{subject to} \quad y^{(1)}_{ti} \le y_{ti} \le C_i - I_{ti}, \;\; i = 1,\dots,N.
\end{aligned} \tag{4.1}$$

Problems (3.21) and (4.1) are equivalent, as we have deducted a constant term from the former problem to reach to the latter one. Let us define:

$$\begin{aligned}
&g(y_{ti}) = f_{ti}((x_1,x_2,\dots,x_{L-1})_{ti}, y^*_{ti}) + F_{ti} \cdot (\mathbb{1}(y^*_{ti} > 0) + F_{t+1,i} \cdot \mathbb{1}((\tilde{y}_{t+1,i}|y^*_{ti}) > 0) \\
&- f_{ti}((x_1,x_2,\dots,x_{L-1})_{ti}, y_{ti}) - F_{ti} \cdot \mathbb{1}(y_{ti} > 0) - F_{t+1,i} \cdot \mathbb{1}((\tilde{y}_{t+1,i}|y_{ti}) > 0)
\end{aligned} \tag{4.2}$$

Then, optimizing Problem (4.1) is equivalent to optimizing Problem (4.3) formulated as follows:

$$\begin{aligned}
&\min_{(y_{t1},\dots,y_{tN})} R(y_{t1},\dots,y_{tN}) + R((\tilde{y}_{t+1,1}|y_{t1}),\dots,(\tilde{y}_{t+1,N}|y_{tN})) + \sum_{i=1}^{N} g(y_{it}) \\
&\text{subject to} \quad y^{(1)}_{ti} \le y_{ti} \le C_i - I_{ti}, \;\; i = 1,\dots,N.
\end{aligned} \tag{4.3}$$

In this chapter, we are going to regard $(y^*_{t1},\dots,y^*_{tN})$ and $((\tilde{y}_{t+1,1}|y^*_{t1}),\dots,(\tilde{y}_{t+1,N}|y^*_{tN}))$ as the initial orders placed by the stores for periods t and $t+1$, respectively. Interpret $g(y_{ti})$ as a penalty if quantities y_{ti} and $(\tilde{y}_{t+1,i}|y_{ti})$ are delivered to store i in periods 1 and 2, respectively, rather than the initial orders of y^*_{ti} and $(\tilde{y}_{t+1,i}|y^*_{ti})$. Note that $g(y^*_{ti}) = 0$, which implies that there is no penalty if the initial orders are delivered. Problem (4.3) is indeed a two-period VRP with penalized delivery shifts.

In this chapter, we discuss the two-period VRP with full delivery shifts. We have already shown that the two-period VRP provides a model with an exact solution method, under some restrictive conditions, to solve Problem (3.21). In the sequel, we introduce the two-period VRP and motivate it as an independent problem to model and solve to optimality a DMPVRP with a rolling horizon of two periods.

4.1.2 Motivation 2: from the DMPVRP to the two-period VRP

Our two-period VRP may be regarded as a chunk of a DMPVRP with a finite/infinite horizon. Consider the planning process presented in Figure 4.1 where each store records stochastic demands from its own customers during period t. Every store i has its own inventory control system whereby, at the end of period t, it individually calculates the optimal orders to be placed for periods $t+1$ and $t+2$. Calculation of such deterministic orders could be based on the current inventory level in store i, on the demand distribution functions of the end customers in periods $t+1$ onward, and on other relevant parameters. But it does not explicitly take into account the global routing costs. The stores have long-term contracts with the LSP. The contracts bind the LSP to serve each store against some pre-agreed annual payment.

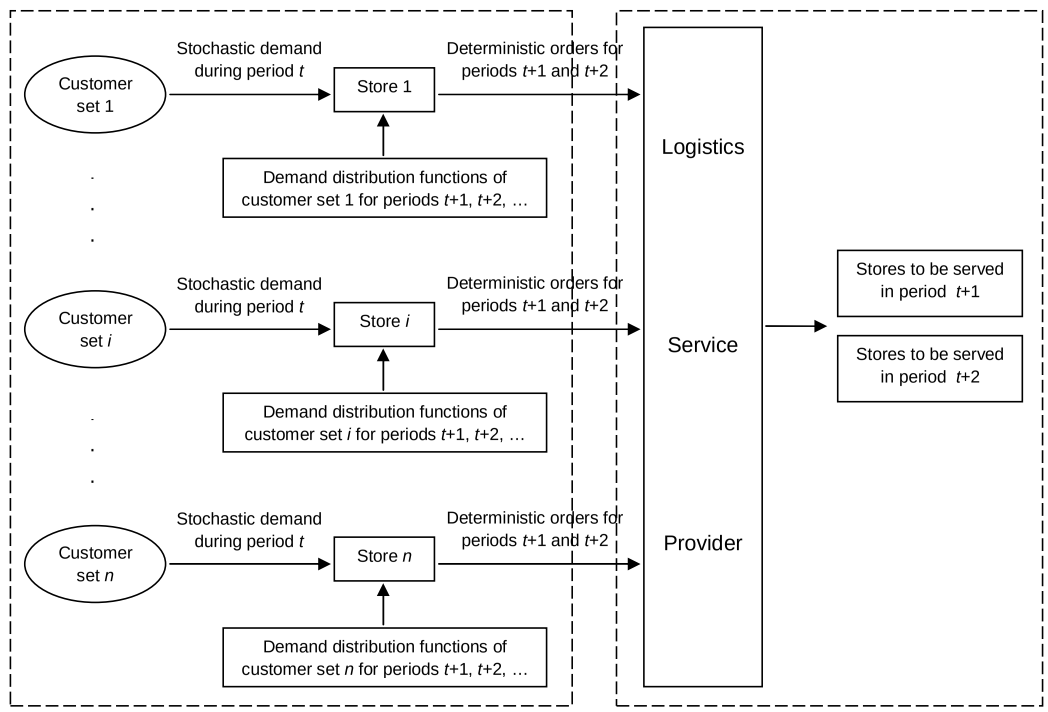


Figure 4.1: The inventory and delivery planning process

In its day-to-day operations, the LSP focuses on two periods only. At the end of period t, the LSP receives the order sizes of the stores for periods $t+1$ and $t+2$, decides about each order whether it should be satisfied or shifted, executes its decision for period $t+1$, and waits until the end of period $t+1$ when orders for periods $t+2$ and $t+3$ are placed by the stores. In other words, a rolling horizon of two periods is considered where the decision for period $t+1$ is executed but the decision for period $t+2$ may still undergo changes. The reason to consider such a short rolling horizon is that demand from the end customers to the stores is stochastic. As a consequence, the "optimal" order quantities computed by each store for periods $t+2$ and beyond may be poorly estimated in period t, especially when the variance of demand is very high. Note that, from a store's point of view, advancement is tantamount to holding unnecessary inventory, and postponement potentially yields lost sales and low service levels. Therefore, shifting by no more than one period is justified when the holding costs are significant, and when the stores are committed to providing very high service levels to their customers.

A typical application with the aforementioned characteristics is inventory control of fresh products in supermarkets, where the products rapidly lose their quality and the stores aim at providing a very high service level. As a result, shifting deliveries by more than one day is undesirable from the stores' point of view. Van Donselaar et al. [2006] and Van Donselaar et al. [2010] conducted an analysis of two Dutch supermarket chains and reported that the average delivery frequency of fresh products to each store is 1.2 days. This is consistent with the results of our interview with the supply chain manager of a Belgian supermarket chain, who confirmed that most of the stores require to be served every day or every other day. Furthermore, the numerical results by Wen et al. [2010] show that, when solving a DMPVRP, considering a rolling horizon of two periods may yield better costs compared to a rolling horizon of either one period or an infinite number of periods. Coelho et al. [2012c] analyze the impact of increasing the length of planning horizon from 3 periods to 6 periods in a dynamic and stochastic inventory routing problem. The results show that solution quality deteriorates with a longer horizon.

By confining ourselves to two periods, the planning process presented in Figure 4.1 can be decomposed into n independent inventory control problems on the left side and a two-period VRP on the right side. This chapter is only dedicated to modeling and to solving the latter problem. Hence, we consider the deterministic orders placed by each store for periods $t+1$ and $t+2$ (or, for simplicity, periods 1 and 2), and we build our model from the LSP's perspective. Although our two-period model focuses on one aspect of the broader DMPVRP, it can be exploited to solve the DMPVRP over a rolling horizon.

4.1.3 Additional discussion

To put our problem in a more formal framework, consider a two-period VRP where deterministic orders of stores are known for two periods. Unlimited supply of the product exists in a warehouse (depot) where the LSP can load an unlimited number of homogeneous capacitated vehicles in each period before delivering the orders to the stores. If the LSP has to satisfy each order in its associated period, then the two-period VRP simplifies into two independent VRPs. But, as we stated before, the LSP may decide to postpone (advance) deliveries for period 1 (period 2), so as to benefit from a decrease in the total routing costs.

We assume that, for each store order in period p ($p = 1$ or 2), either the order is *fully* delivered in period p, or no delivery takes place in period p and a delivery must be performed in the other (next or previous) period. (We do not discuss partial shifts, although this may be regarded as a natural variant of our problem.) A crucial feature of our model is that *the size of an order can change when its delivery is shifted from one period to another one*. To explain this feature, let us assume that the order sizes in periods 1 and 2 for some store i are equal to 4 units and 0 unit, respectively. Depending on its inventory control system, in case its initial order size for period 1 is postponed, store i may choose to receive an order size different from 4 units in period 2, e.g., 2 units or 5 units. This may be justified, in particular, if the demand of period 1 cannot be backlogged.

There is an a priori defined penalty associated with each shift of deliveries, which may depend on the magnitude of the order. Figure 4.2 shows a solution of the two-period VRP where orders of the stores are satisfied in their associated periods, i.e., neither postponements nor advancements take place. In this figure, each store is represented by a circle, and the depot is depicted by a triangle. The quantity above each vertex (store) represents its initial order, and the capacity of each vehicle is taken to be 10 units.

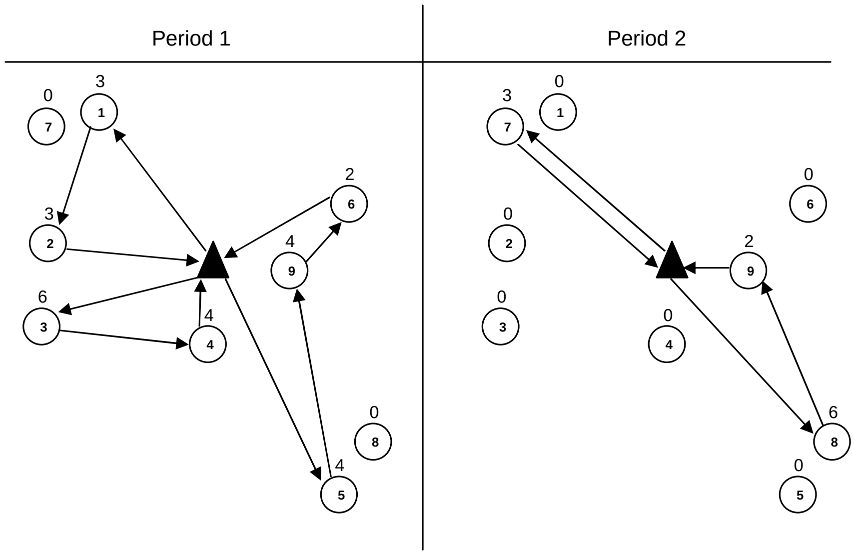


Figure 4.2: Optimal routes when shifts are not allowed

Assume that when shifting deliveries, the shifted quantity requested by store 5 in period 2 is 2 units, and the shifted quantity requested by store 7 in period 1 is 4 units. Visually, it is easy to imagine that, in Figure 4.2, we can decrease the routing costs in period 1 by postponing the delivery to store 5, while just slightly adding to the routing costs in period 2 by inserting store 5 into route $0-8-9-0$. Simultaneously, we can further decrease the routing costs in period 2 by advancing the delivery to store 7 and by inserting it into route $0-1-2-0$ in period 1. The new solution where demands of store 5 and store 7 are postponed and advanced, respectively, is shown in Figure 4.3. In this figure, we have to take into consideration two penalties, namely, the penalty of postponing the order of store 5 and the penalty of advancing the order of store 7.

Whether the new solution in Figure 4.3 is better than the solution in Figure 4.2 depends on how much the LSP saves in the routing costs and how much it has to pay for the penalties.

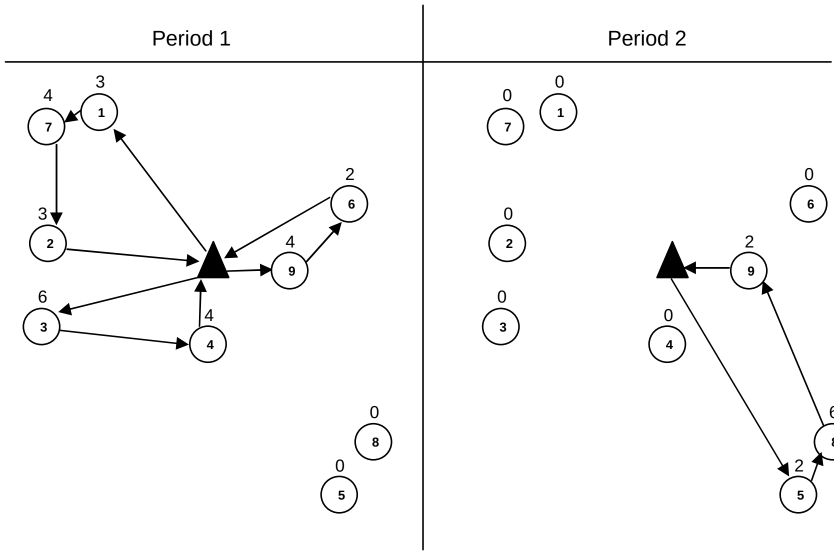


Figure 4.3: Optimal routes when shifts are allowed

4.1.4 Scientific contributions

The main contributions of this chapter can be summarized as follows:

- We formulate the two-period VRP as an Integer Linear Programming (ILP) problem. Unlike the existing deterministic models in the multi-period VRP literature (see Section 4.1.5), our model can deal with the case where the sum of the orders of a store for two periods is not a fixed quantity.
- We solve the model to optimality by a branch-and-price algorithm. We introduce two new acceleration techniques derived from the structure of our problem to improve the speed of the label-setting algorithm. The techniques have proved efficient in our experiments.
- We draw algorithmic insights by solving an ILP model with a restricted number of generated columns in each node, and we analyze the trade-off between the computational time and the optimality gap.
- We draw managerial insights on cost improvements based on the results obtained from the test instances. In particular, we underline the benefits obtained by solving the integrated two-period VRP model, rather than two independent VRP models.

4.1.5 Related works

Our two-period VRP shares similarities with the PVRP, the MPVRP, and the IRP, but also has some differences. In the PVRP, a total demand of W_i units must be delivered to customer $i$during the planning horizon. One of the feasible delivery schedules must be assigned to each customer. Our two-period VRP looks like a PVRP with two periods, where there are two or three delivery schedules for each customer. However, in a PVRP delivery quantities are fixed while in our problem the delivery quantities for different schedules (with the same delivery frequency) of a customer can be different.

The IRP models a centralized decision-making system where a common vendor is responsible for replenishing a set of stores so that the stores do not run out of stock. In the IRP, the required delivery quantities can be delivered in advance but cannot be postponed, whereas in the two-period VRP we have defined they can be advanced, postponed, and change in quantity.

In an MPVRP, a single order is placed by each store which must be served within a *period window*, i.e., a set of consecutive periods in the planning horizon (see, e.g., Archetti et al. 2015a). In other words, each order has a release date and a due date, which represent, respectively, the earliest period and the latest period in which the order can be delivered. An LSP decides on the delivery period of each order and the delivery routes in each period, but the delivery quantities are fixed and cannot be split. The two-period VRP we have defined is not an MPVRP, as it contains stores with more than one requested delivery over the planning horizon. If our problem did not contain any customers with positive demands in both periods and delivery quantities did not change when shifting, it would be an MPVRP where release date is 1 and due date is 2 for all customers, holding cost is positive for customers with a positive demand only for period 1, and is negative for customers with a positive demand only for period 2.

Archetti et al. [2015a] investigate a more general MPVRP, where the formulation includes not only regular stores with due dates within the planning horizon, but also a set of stores with due dates beyond it. The model allows for postponing deliveries to the latter stores until some (undetermined) period beyond the planning horizon, at a cost (this amounts to skipping deliveries to these stores). The LSP incurs an inventory holding cost for the orders which are delivered after their release dates. This holding cost behaves like a postponement penalty for each period of delay after the release date. The assumption of having the stores which may completely be skipped at a penalty can easily be included in our model. [Archetti et al., 2015a] propose that multiple orders of the same customer can be modeled through different co-located customers. However, in this way of modeling, the orders from co-located customers are not necessarily delivered by the same vehicle, i.e., the assumption of at most one visit to each customer in each period is not guaranteed.

Albareda-Sambola et al. [2014] consider a DMPVRP, where at the end of each period, exact information about the orders placed in that period and earlier periods is available, and partial information about the orders upcoming in subsequent periods is gradually revealed. Albareda-Sambola et al. [2014] develop a formula to measure the approximate profit of serving each store in the current period. In order to decide which stores should be served in the current period, they formulate a VRP where the objective function accounts for profit collection and routing costs.

Angelelli et al. [2007a,b] handle a DMPVRP with a single uncapacitated vehicle where, in each period, a set of orders appear. The release date for orders is 1 and the due date is either 1 or 2. The orders with a due date in period 2 can be delivered in any period (say, “advanced” to period 1 or “postponed” to period 2) without penalty. The authors consider a planning horizon of two periods and they analyze the competitive ratios of three simple heuristics to determine which orders should be delivered in period 1. The heuristics are: (a) deliver orders with due date 1, (b) deliver orders with due date 1 or 2, and (c) deliver orders with due date 1 and a subset of orders with due date 2 which are located close to the orders with due date 1. Angelelli et al. [2009] analyze a similar problem and develop a variable neighborhood search heuristic to solve it. Contrary to Angelelli et al. [2007a,b] where the planning horizon is restricted to two periods, Angelelli et al. [2009] consider a longer planning horizon and analyze the impact of short term strategies on the average daily operational costs.

To the best of our knowledge, the two-period VRP model with penalized shifts, as we define it in this chapter, has not been previously examined in the literature. In the planning process represented in Figure 4.1, we assume an RMI-like system where each store has its own inventory control system and determines its orders based on the demand from its end-customers. Thus, we consider a DMPVRP from the LSP's perspective. We model and solve only a fragment of this problem, that is, a two-period VRP. In this sense, our problem is a special MPVRP with deterministic orders placed for two periods and a planning horizon of two periods. However, we assume that the LSP has flexibility to choose the delivery periods, i.e., deliveries can be advanced or postponed by one period but are penalized. By contrast, whereas the existing models on the MPVRP consider a single fixed order for each store for the entire planning horizon, in our model each store is allowed to place orders for two periods and the sum of the orders for two periods is not necessarily a fixed quantity. Indeed, when a delivery is shifted, the corresponding store may require a different delivery quantity. In any case, the quantity of a shifted delivery is determined by the corresponding store and not by the LSP.

The remainder of chapter is organized as follows. The problem statement is presented in Section 4.2. The column generation process, including the label-setting algorithm for the solution of two pricing sub-problems, is discussed in Section 4.3. We deal with details of implementation issues in Section 4.4. Computational results including algorithmic and managerial insights are presented in Section 4.5, and finally, conclusions are drawn in Section 4.6.

4.2 Problem statement

We use the notations in Tables 4.1-4.3. For the sake of convenience, we will redefine some of them in the course of our discussion.

Consider a graph $G = (V^0, A)$ where vertices represent the depot (denoted by 0) and the stores, and arcs represent transportation links. The travel times t_{ij} satisfy the triangle inequalities. Products are picked up from the depot and delivered to the stores within given time windows. Each route starts and ends at the depot. The total transportation costs over two periods, including the penalties, should be minimized. The LSP has access to an unlimited homogeneous fleet with the same capacity Q for each vehicle. A variable cost equal to c_{ij} is incurred when arc (i, j) is traversed. Each vehicle can perform at most one single route per period. Split deliveries within a period are not allowed, that is, each store is served by at most one vehicle in each period.

For the sake of simplicity in our notations, we denote the current period t by 0. Define d_{i1} and d_{i2} as the orders of store i for periods 1 and 2, respectively. Without loss of generality, we assume that each store has a positive order in at least one period.

Table 4.1: Indices and sets

i, j	indices for vertices (stores)
r	index for routes
V_{+0}	set of stores with a positive order for period 1 and no order for period 2
V_{0+}	set of stores with no order for period 1 and a positive order for period 2
V_{++}	set of stores with positive orders for both periods
V'_{++}	set of virtual stores associated with the real stores in V_{++}
V	$V_{+0} \cup V_{0+} \cup V_{++} \cup V'_{++}$
V^0	$V \cup \{0\}$ where vertex 0 denotes the depot
A	set of arcs
R_1	set of feasible routes in period 1
R_2	set of feasible routes in period 2

As in Figure 4.2, we can distinguish three classes of stores. Any store i in class V_{+0} has a positive order d_{i1} for period 1 but no order for period 2. If the LSP decides not to deliver to store i in period 1 (postponing), then it is required to deliver d'_{i2} units in period 2, where the quantity d'_{i2} is set by store i (d'_{i2} might be smaller than, equal to, or greater than d_{i1}). However, the LSP is charged a penalty p_i for making this alternative decision (postponing). Similarly, a store i in class V_{0+} has no order for period 1 but a positive order d_{i2} for period 2. The alternative decision (advancing) for the LSP is to deliver d'_{i1} units in period 1 and zero units in period 2. Here again, d'_{i1} is set by the store and might be different from d_{i2}. The LSP is charged a penalty a_i for advancing the order. Finally, class V_{++} includes stores with positive orders d_{i1} and d_{i2} for both periods. In this class, two alternative decisions can be made for each store. The first alternative decision (postponing) is to deliver zero unit in period 1 and d'_{i2} units in period 2. The second alternative decision (advancing) is to deliver d'_{i1} in period 1 and zero unit in period 2. Neither d'_{i1} nor d'_{i2} need to be equal to $d_{i1} + d_{i2}$. Penalties p_i and a_i are incurred for postponing and advancing, respectively. Table 4.3 summarizes all possible decisions regarding delivery quantities for the three classes. If either p_i or a_i is infinite, we say that the corresponding shift is *forbidden*.

The two-period VRP is obviously NP-hard since it generalizes the classical VRP. An integer linear programming (ILP) formulation of the two-period VRP can be obtained, as for the VRP, by introducing decision variables corresponding to the selection of feasible routes in each period; see Table 4.4, where R_1 and R_2 denote the sets of feasible routes in periods 1 and 2, respectively.

However, in this formulation, a difficulty arises with the interpretation of the deliveries. To understand this difficulty, note first that, when some store $i \in V_{+0}$ is visited by a route in period 1, it means that the initial orders are delivered to this store in periods 1 and 2 (namely, d_{i1} and zero units, respectively). On the other hand, if the LSP decides to serve store $i \in V_{+0}$ in period 2, then it has necessarily made the alternative decision with delivery quantities zero and d'_{i2} in periods 1 and 2, respectively. A similar reasoning applies for stores $i \in V_{0+}$.

On the other hand, this reasoning fails for a store $i \in V_{++}$. Indeed, if store $i \in V_{++}$ is visited in period 1 (i.e., $u_{r1} = 1$ for some route r containing i), then the size of the delivery quantities to this store in period 1 and in period 2 is not immediately known.

Table 4.2: Parameters

d_{i1}	order of store i for period 1
d_{i2}	order of store i for period 2
d'_{i1}	order of store $i \in V_{0+} \cup V_{++}$ for period 1 if it is not served in period 2
d'_{i2}	order of store $i \in V_{+0} \cup V_{++}$ for period 2 if it is not served in period 1
p_i	postponement penalty imposed by store i
a_i	advancement penalty imposed by store i
n_1	number of stores in set V_{+0}
n_2	number of stores in set V_{0+}
n_3	number of stores in set V_{++}
n	total number of stores including virtual stores $(n = n_1 + n_2 + 2n_3)$
c_{ij}	cost of using arc (i, j)
Q	capacity of each vehicle
t_{ij}	travel time to traverse arc (i, j)
s_i	service time at store i
(e_i, l_i)	time window for the arrival of a vehicle at vertex i
α_{ir}	1 if store i belongs to route r; 0 otherwise.

Table 4.3: Three classes of stores: feasible decisions, delivery quantities, and associated penalties

	Initial decision			Alt. decision 1			Alt. decision 2		
Class	$del._1$	$del._2$	$pe.$	$del._1$	$del._2$	$pe.$	$del._1$	$del._2$	$pe.$
V_{+0}	d_{i1}	0	0	0	d'_{i2}	p_i	—	—	—
V_{0+}	0	d_{i2}	0	d'_{i1}	0	a_i	—	—	—
V_{++}	d_{i1}	d_{i2}	0	0	d'_{i2}	p_i	d'_{i1}	0	a_i

It only appears, from Table 4.3, that the LSP has made either the initial decision with delivery quantities d_{i1} and d_{i2} in periods 1 and 2, or the second alternative decision with delivery quantities d'_{i1} and zero in periods 1 and 2. Similarly, if some store $i \in V_{++}$ is served in period 2, then it implies that the LSP has made either the initial decision with delivery quantities d_{i1} and d_{i2} in periods 1 and 2, or the first alternative decision with delivery quantities zero and d'_{i2} in periods 1 and 2, respectively.

To resolve this ambiguity, we assume from now on that if store $i \in V_{++}$ is visited by some route r in period 1 (respectively, period 2), then the delivery quantity in period 1 (respectively, period 2) is d_{i1} (respectively, d_{i2}). Furthermore, we define a *virtual store* $i + n_3 \in V'_{++}$ corresponding to store $i \in V_{++}$ with orders $(d'_{i1} - d_{i1})$ and $(d'_{i2} - d_{i2})$ for periods 1 and 2, respectively; these quantities could be negative.

The cost of traveling to or from a virtual store $i + n_3$ is set in such a way that $i + n_3$ can be visited only if the associated real store i is visited in the same period and on the same route, immediately before $i + n_3$. In this way, visiting i and $i + n_3$ on the same route in period 1 (respectively, 2) means that the alternative decision 2 (respectively, 1) has been made for store $i \in V_{++}$. If the virtual store $i + n_3$ is visited in neither period, our interpretation is that the LSP delivers the initial orders d_{i1} and d_{i2}

Table 4.4: Decision variables

$u_{r1} = 1$	if route $r \in R_1$ is used in period 1; 0 otherwise.
$u_{r2} = 1$	if route $r \in R_2$ is used in period 2; 0 otherwise.

to store i. This information is summarized in Table 4.5, where del_{i1} and del_{i2} denote the nonzero delivery quantities to store i when this store is visited in period 1 and in period 2, respectively.

Table 4.5: Delivery quantities to store i in period 1 and in period 2

Class	del_{i1}	del_{i2}
$i \in V_{+0}$	d_{i1}	d'_{i2}
$i \in V_{0+}$	d'_{i1}	d_{i2}
$i \in V_{++}$	d_{i1}	d_{i2}
$i \in V'_{++}$	$(d'_{i-n_3,1} - d_{i-n_3,1})$	$(d'_{i-n_3,2} - d_{i-n_3,2})$

More specifically, the cost of traveling from real stores to virtual stores is expressed by Equation (4.4). The cost of traveling from virtual stores to real stores is given by Equation (4.5).

$$c_{ij} = \begin{cases} 0 & j \in V'_{++},\ j = i + n_3, \\ \infty & j \in V'_{++},\ j \neq i + n_3, \end{cases} \tag{4.4}$$

$$c_{ij} = c_{i-n_3,j} \quad i \in V'_{++},\ j \in V_{+0} \cup V_{0+} \cup V_{++}. \tag{4.5}$$

4.3 Column generation

This section discusses an Integer Linear Programming (ILP) formulation of the two-period VRP. We develop a column generation algorithm to solve the LP-relaxation of the ILP, and we describe in detail a label-setting algorithm to solve the pricing problem raised from the LP-relaxation. We formulate two pricing subproblems that generate feasible routes in each period. The pricing problems are Elementary Shortest Path Problems with Resource Constraints (ESPPRC). We explain in Sections 4.3.4-4.3.6 how they can be solved efficiently.

4.3.1 An integer linear programming formulation

A route is feasible if (1) the total delivery quantity in it does not exceed the vehicle capacity, (2) it respects the time windows, (3) it starts and ends at the depot and visits each vertex at most once (elementarity), (4) it does not include forbidden shifts. Note that if a route includes any virtual vertices, they will immediately succeed their corresponding real vertices due to Equation (4.4). We can now formulate the two-period VRP as an ILP problem, as follows.

$$
\begin{aligned}
\min & \sum_{r\in R_1}\sum_{(i,j)\in r} c_{ij}u_{r1} + \sum_{r\in R_2}\sum_{(i,j)\in r} c_{ij}u_{r2} \\
& + \sum_{i\in V_{+0}} p_i\Big(\sum_{r\in R_2}\alpha_{ir}u_{r2}\Big) + \sum_{i\in V_{0+}} a_i\Big(\sum_{r\in R_1}\alpha_{ir}u_{r1}\Big) \\
& + \sum_{i\in V_{++}} p_i\Big(\sum_{r\in R_2}\alpha_{i+n_3,r}u_{r2}\Big) + \sum_{i\in V_{++}} a_i\Big(\sum_{r\in R_1}\alpha_{i+n_3,r}u_{r1}\Big)
\end{aligned} \tag{4.6}
$$

subject to

$$\sum_{r\in R_1}\alpha_{ir}u_{r1} + \sum_{r\in R_2}\alpha_{ir}u_{r2} = 1;\ \forall i\in V_{+0},\ (\text{dual variables: } \beta_i) \tag{4.7}$$

$$\sum_{r\in R_1}\alpha_{ir}u_{r1} + \sum_{r\in R_2}\alpha_{ir}u_{r2} = 1;\ \forall i\in V_{0+},\ (\text{dual variables: } \gamma_i) \tag{4.8}$$

$$\sum_{r\in R_1}\alpha_{ir}u_{r1} + \sum_{r\in R_2}\alpha_{i+n_3,r}u_{r2} = 1;\ \forall i\in V_{++},\ (\text{dual variables: } \lambda_i) \tag{4.9}$$

$$\sum_{r\in R_1}\alpha_{i+n_3,r}u_{r1} + \sum_{r\in R_2}\alpha_{ir}u_{r2} = 1;\ \forall i\in V_{++},\ (\text{dual variables: } \mu_i) \tag{4.10}$$

$$u_{r1}\in\{0,1\};\ \forall r\in R_1 \tag{4.11}$$

$$u_{r2}\in\{0,1\};\ \forall r\in R_2 \tag{4.12}$$

The objective function (4.6) consists of variable costs of each route in both periods, postponement penalty for any store in class V_{+0} if it is included in a route selected in period 2, advancement penalty for any store in class V_{0+} if it is included in a route selected in period 1, postponement penalty for any store in class V_{++} if its corresponding virtual store is included in any route selected in period 2, and advancement penalty for any store in class V_{++} if its corresponding virtual store is included in a route selected in period 1. Constraints (4.7) and (4.8) guarantee that every store in $V_{+0}\cup V_{0+}$ is served either in period 1 or in period 2. Constraints (4.9) impose that if any store in class V_{++} is served in period 1, then its associated virtual store is not served in period 2 (no postponement of delivery), and conversely. Constraints (4.10) are interpreted similarly: if a store in V_{++} is served in period 2, then its associated virtual store is not served in period 1 (no advancement of delivery), and conversely.

4.3.2 Master problem

The LP-relaxation of problem (4.6)-(4.12) is viewed as a master problem that can be solved by column generation [Lübbecke and Desrosiers, 2005; Dabia et al., 2013]. This approach allows us to deal implicitly with various constraints of the problem, like time windows or vehicle capacity constraints, which are not explicitly mentioned in the formulation (4.6)-(4.12). We could similarly handle additional side constraints, like forbidding delivery shifts for certain stores or the position of virtual stores in the routes. Note that all these constraints are already included in the definition of a feasible route.

4.3.3 Pricing problems

In order to solve the master problem by column generation, we formulate a pricing problem for each period as an ESPPRC [Irnich and Desaulniers, 2005]. Each feasible solution of the ESPPRC is a route which starts and ends at the depot while including a subset of the vertices and respecting the side constraints related to the vehicle load, time windows, and permissible shifts. The settings are done in such a way that the cost of a route (solution) in the ESPPRC is equal to the reduced cost of the same route in the master problem.

More precisely, based on the dual prices obtained in the optimal solution of the restricted master problem, the reduced cost of a route r in period 1 is calculated as $\sum_{(i,j)\in r} \bar{c}_{ij}$, where the cost coefficients $\bar{c}_{ij}$'s for period 1 are defined by Equation (4.13), where we define $\beta_0 = 0$ by convention:

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \beta_j; & \forall i \in V^0,\ j \in V^0_{+0} \backslash \{i\} \\ c_{ij} + a_j - \gamma_j; & \forall i \in V^0,\ j \in V_{0+} \backslash \{i\} \\ c_{ij} - \lambda_j; & \forall i \in V^0,\ j \in V_{++} \backslash \{i\} \\ c_{ij} + a_{j-n_3} - \mu_{j-n_3}; & \forall i \in V^0,\ j \in V'_{++} \backslash \{i\} \end{cases} \tag{4.13}$$

The reduced cost of a route in period 2 is similarly calculated, with the cost coefficients $\bar{c}_{ij}$'s for period 2 defined by Equation (4.14), with $p_0 = 0$ and $\beta_0 = 0$ by convention:

$$\bar{c}_{ij} = \begin{cases} c_{ij} + p_j - \beta_j; & \forall i \in V^0,\ j \in V^0_{+0} \backslash \{i\} \\ c_{ij} - \gamma_j; & \forall i \in V^0,\ j \in V_{0+} \backslash \{i\} \\ c_{ij} - \mu_j; & \forall i \in V^0,\ j \in V_{++} \backslash \{i\} \\ c_{ij} + p_{j-n_3} - \lambda_{j-n_3}; & \forall i \in V^0,\ j \in V'_{++} \backslash \{i\} \end{cases} \tag{4.14}$$

For each period 1 and 2, we set up a distinct network on the vertex set V^0, where the cost $\bar{c}_{ij}$ of each arc (i,j) is given either by (4.13) or by (4.14), depending on the period. Other parameters of the network are listed in Table 4.2. In each network, we seek the feasible routes with negative cost, i.e., the feasible routes with negative reduced cost in the master problem. Feasibility of a route in the ESPPRC implies that it is feasible in the master problem, too. As long as $\min \sum_{(i,j)\in r} \bar{c}_{ij}$ in either period is negative there exists a route which is potentially able to improve the objective function of the master problem. More generally, any solution of the ESPPRC in period 1 or 2 with a negative cost (not necessarily the optimal solution) can be introduced in the master problem in the next iteration. Column generation stops when neither the ESPPRC in period 1 nor the ESPPRC in period 2 is able to identify a route with negative cost.

4.3.4 The label-setting algorithm

A label-setting algorithm is used to solve the pricing problems (ESPPRC). In this algorithm, a multi-dimensional label L_i is associated with each path from the depot to an end vertex i. In expanded form, the components of label L_i are $(L_i^{cost}, L_i^{load}, L_i^{time}, (L_i^k)_{k\in V})$, where each component indicates the consumption of a limited resource along the path with which L_i is associated. The path under consideration is feasible if all components of L_i respect the limits on available resources.

The first component L_i^{cost} denotes the sum of $\bar{c}_{ij}$'s over the arcs (i,j) covered by the path, and there is no resource constraint on it. The second component L_i^{load} is the sum of the delivery quantities to all vertices visited by the path, including vertex i. When generating a path, the load on the corresponding vehicle should not exceed the vehicle capacity Q, that is,

$$0 \leq L_i^{load} \leq Q. \tag{4.15}$$

The third component L_i^{time} shows the time when store i is visited and the service starts at this store. It must respect the time window for store i, that is,

$$e_i \leq L_i^{time} \leq l_i. \tag{4.16}$$

Finally, L_i^k indicates the number of times store k is visited by path L_i. Each path must be elementary, meaning that a path cannot visit any store k more than once:

$$0 \leq L_i^k \leq 1 \quad \text{for all } k \in V. \tag{4.17}$$

The label-setting algorithm starts from the initial label $L_0 = (0, 0, e_0, (0)_{k \in V})$, associated with the depot, and generates new labels using extension functions. A label L_i is extended along all arcs $(i, j) \in A$ and new labels $L_j = (L_j^{cost}, L_j^{load}, L_j^{time}, (L_j^k)_{k \in V})$ are created, where:

$L_j^{cost} = L_i^{cost} + \bar{c}_{ij}$,
$L_j^{load} = L_i^{load} + del_{jt}$ (where t is the period under consideration and del_{jt} is as in Table 4.5),
$L_j^{time} = \max\{e_j, L_i^{time} + s_i + t_{ij}\}$,

$$L_j^k = \begin{cases} L_j^k + 1 & \text{if } k = j \\ L_j^k & \text{otherwise.} \end{cases}$$

A label L_j is discarded if at least one of its resource components exceeds the corresponding limits in inequalities (4.15)-(4.17). A feasible route is constructed by extending a feasible path to the depot, provided that the extended label to the depot remains feasible.

To avoid enumerating all feasible paths, a dominance rule is applied to eliminate labels that are not Pareto optimal and, therefore, cannot yield an optimal path [Gutierrez-Jarpa et al., 2010]. Given two labels $(L_j)_1 = (L_j^{cost}, L_j^{load}, L_j^{time}, (L_j^k)_{k \in V})_1$ and $(L_j)_2 = (L_j^{cost}, L_j^{load}, L_j^{time}, (L_j^k)_{k \in V})_2$ ending at the same vertex j, this rule stipulates that $(L_j)_1$ dominates $(L_j)_2$ if $(L_j)_1 \leq (L_j)_2$ component-wise and the inequality is strict for at least one component.

The basic version of the label-setting algorithm is not very efficient [Irnich and Desaulniers, 2005], but various techniques can be used to speed it up. We use three classical acceleration techniques from the literature. Moreover, we introduce two new techniques derived from the structure of our problem which have proved efficient in our experiments.

4.3.5 Classical acceleration techniques

4.3.5.1 Bidirectional search.

Righini and Salani [2006] have proposed a bounded bi-directional search algorithm, where labels are extended from the depot to other vertices not only forwardly but also backwardly. In the first step, as in the mono-directional algorithm, labels are extended forwardly from the depot as long as the time component L_i^{time} does not exceed half of the available driving time of each vehicle; that is, as long as $L_i^{time} < 0.5(l_0 - e_0)$. The second step deals with extending labels backwardly from the depot using backward extension functions. Denote by $\Gamma_j = (\Gamma_j^{cost}, \Gamma_j^{load}, \Gamma_j^{time}, (\Gamma_j^k)_{k \in V})$ a backward label associated with a partial path $(j, \ldots, 0)$. The time component in this backward label is denoted as Γ_i^{time}; it indicates the latest time at which a vehicle can reach store i.

The backward labeling step starts with an initial label $\Gamma_0 = (0, 0, l_0, (0)_{k \in V})$. Each non-dominated label Γ_j is extended backwardly along all arcs $(i, j) \in A$ using the following resource extension functions, as long as $\Gamma_i^{time} > 0.5(l_0 - e_0)$:

$\Gamma_i^{cost} = \Gamma_j^{cost} + \bar{c}_{ij}$,
$\Gamma_i^{load} = \Gamma_j^{load} + del_{it}$ (where t is the period under consideration and del_{it} is as in Table 4.5),
$\Gamma_i^{time} = \min\{l_i, \Gamma_j^{time} - t_{ij} - s_i\}$,

$$\Gamma_i^k = \begin{cases} \Gamma_i^k + 1 & \text{if } k = i \\ \Gamma_i^k & \text{otherwise.} \end{cases}$$

The label created at vertex i is discarded if $\Gamma_i^{time} \leq 0.5(l_0 - e_0)$, or $\Gamma_i^{load} > Q$, or $\Gamma_i^{time} < e_i$, or $\Gamma_i^k > 1$ for at least one k. The dominance rule is adapted as follows. Let $(\Gamma_i)_1 = (\Gamma_i^{cost}, \Gamma_i^{load}, \Gamma_i^{time}, (\Gamma_i^k)_{k \in V})_1$ and $(\Gamma_i)_2 = (\Gamma_i^{cost}, \Gamma_i^{load}, \Gamma_i^{time}, (\Gamma_i^k)_{k \in V})_2$ be two labels representing partial paths starting at the same vertex i. Then, $(\Gamma_i)_1$ dominates $(\Gamma_i)_2$ if $(\Gamma_i)_1 \leq (\Gamma_i)_2$ for all components except the time component, $(\Gamma_i^{time})_1 \geq (\Gamma_i^{time})_2$, and the inequality is strict for at least one component.

Finally, in a third step, pairs of forward and backward labels associated with the same vertex are joined together in order to build complete routes starting and ending at the depot: If L_i is a forward label associated with a path $(0, \ldots, i)$ ending at vertex i, and Γ_j is a backward label associated with a path $(j, \ldots, 0)$ starting from vertex j, then concatenating the two paths yields a complete route $(0, \ldots, 0)$ with cost $L_i^{cost} + \Gamma_j^{cost} + \bar{c}_{ij}$. The route is feasible if the following conditions hold:

$L_i^{load} + \Gamma_j^{load} \leq Q$,
$L_i^{time} + s_i + t_{ij} \leq \Gamma_j^{time}$,
$L_i^k + \Gamma_j^k \leq 1$ for all $k \in V$.

4.3.5.2 Inaccessible vertices.

This technique was proposed by Feillet et al. [2004]. It suggests to set the value of L_i^k (resp., Γ_j^k) to 1 not only when store k has been visited along the path associated with L_i (resp., Γ_j) but also when store k cannot be visited anymore because of the time window constraints. In other words, L_i^k is set to 1 when store k is visited along the path or when $\max\{e_k, L_i^{time} + s_i + t_{ik}\} > \min\{0.5(l_0 - e_0), l_k\}$. For the backward labels, Γ_j^k is set to 1 when store k is visited along the path or when $\min\{l_k, \Gamma_j^{time} - t_{kj} - s_k\} < \max\{0.5(l_0 - e_0), e_k\}$.

4.3.5.3 Relaxed dominance rule.

When comparing two forward labels $(L_i)_1$ and $(L_i)_2$ ending at vertex i, determining whether $(L_i^k)_1 \leq (L_i^k)_2$ for all $k \in V$ can be quite time consuming [Jepsen et al., 2008]. Therefore, we consider the following relaxed version of the dominance rule: we say that $(L_i)_1$ dominates $(L_i)_2$ if $(L_i)_1 \leq (L_i)_2$ for all components but L_i^k $(k \in V)$, and if $\sum_k (L_i^k)_1 \leq \sum_k (L_i^k)_2$. When the latter condition holds, it means that $(L_i)_1$ has visited fewer stores than $(L_i)_2$, although the set of stores visited by $(L_i)_1$ is not necessarily a subset of the stores visited by $(L_i)_2$. Such a relaxation increases the number of dominated labels but may lead to discarding some routes with negative cost even though such routes are actually not dominated. Hence, in each iteration of solving the ESPPRC, we first apply the relaxed dominance rule. If no route with a negative cost is found, we solve again the ESPPRC, this time with the non-relaxed (exact) dominance rule. We apply a similar technique for backward labels.

4.3.6 New acceleration techniques

4.3.6.1 Initial elimination.

Under some circumstances, when the saving in routing costs obtained by postponing a delivery to a store is smaller than the penalty incurred for postponing, we can safely assume that the delivery will take place in period 1, and we can eliminate the (virtual) store from the network in period 2 without losing the optimal solution. A similar conclusion applies for advancing deliveries. Proposition 1 states these observations in a more formal way.

Proposition 1. *For a store $i \in V_{+0} \cup V_{++}$, if $f + c_{0i} + c_{i0} \leq p_i$ and $d_{i2} \leq d'_{i2}$, then there is an optimal solution of the two-period VRP where the delivery to store i is not postponed, and the (virtual) store can be eliminated from the network in period 2. Similarly, if $f + c_{0i} + c_{i0} \leq a_i$ and $d_{i1} \leq d'_{i1}$ for a store $i \in V_{0+} \cup V_{++}$, then there is an optimal solution where the delivery to store i is not advanced and the (virtual) store can be eliminated from the network in period 1.*

Proof. We only handle the case of postponement; the other case is analogous. Consider a store $i \in V_{+0} \cup V_{++}$ and consider any feasible solution where no delivery is made to the store in period 1 and d'_{i2} units are delivered in period 2 (see Table 4.3). Another solution can be obtained by using a single truck to deliver the quantity d_{i1} to store i in period 1, and by delivering d_{i2} units to store i in period 2 without modifying the other routes (if $d_{i2} = 0$, store i should simply be skipped in period 2). The new solution is feasible since $d_{i2} \leq d'_{i2}$. As compared to the original solution, it implies an additional routing cost of $f + c_{0i} + c_{i0}$ in period 1, but it does not incur the penalty p_i. Therefore, if $f + c_{0i} + c_{i0} \leq p_i$, then the total cost of the new solution is not larger than the cost of the original one. This establishes the claim. □

Note that the condition $d_{i2} \leq d'_{i2}$ certainly holds when $i \in V_{+0}$, since $d_{i2} = 0$ in that case. In general, Proposition 1 can be used to reduce the size of the ESPPRC networks to be considered, and this results in significant gains in computing time.

4.3.6.2 Dynamic elimination.

Proposition 2 states that, depending on the dual values λ_i, μ_i which enter the definition of the ESPPRC, some virtual vertices can be eliminated from the current pricing problem without losing its optimal solution. (Note that these virtual vertices may need to be considered again in the next column generation step, depending on the updated values of the dual variables.)

Proposition 2. *Let μ_i and λ_i be the dual variables associated with store $i \in V_{++}$ after solving the LP-relaxation of the master problem. In the pricing problem for period 2 (respectively, period 1), virtual vertex $i+n_3$ can be eliminated if $d_{i2} \leq d'_{i2}$ and $\lambda_i \leq p_i$ (respectively, if $d_{i1} \leq d'_{i1}$ and $\mu_i \leq a_i$).*

Proof. Here we prove the case of period 2; the other case is analogous. Consider first any finite forward label at vertex $i+n_3$, say, L_{i+n_3}. In view of Equation (4.4), it is clear that L_{i+n_3} must have been obtained by extending a forward label L_i computed at vertex i. Let us show that, under the assumptions of the proposition, label L_i dominates L_{i+n_3}, i.e., $L_i \leq L_{i+n_3}$. Observe that L_{i+n_3} includes the same sequence of stores as L_i, plus store $i+n_3$. Therefore,

$L_i^{cost} \leq L_{i+n_3}^{cost}$ since $\bar{c}_{i,i+n_3} = p_i - \lambda_i \geq 0$ by assumption,
$L_i^{load} \leq L_{i+n_3}^{load}$ since $d'_{i2} - d_{i2} \geq 0$ by assumption,
$L_i^{time} = L_{i+n_3}^{time}$,
$L_i^k \leq L_{i+n_3}^k$ for all stores k.

Now, the usual dominance rules do not directly allow us to eliminate L_{i+n_3}, because the paths associated with L_i and L_{i+n_3} end at different stores. Assume, however, that we attempt to extend label L_{i+n_3} forward to any vertex j, thus producing a new label L'_j. In view of Equation (4.5) and of Equation (4.14), it should be clear that label L_i can also be extended to vertex j and will give rise to another label L_j such that $L_j \leq L'_j$, so that L'_j can be eliminated. Thus, it is not worth extending L_{i+n_3} forward, and this label may as well be eliminated immediately.

The proof for backward labels is similar. Consider any backward label Γ_j at vertex j. Assume that Γ_i and Γ_{i+n_3} are two labels derived from Γ_j when extending it backwardly to vertices i and $i+n_3$, respectively. Label Γ_{i+n_3} cannot be finitely extended backward to any vertex but i, in view of Equation (4.4) (by construction, a virtual vertex can only be preceded by its associated real vertex). Let $\Gamma_{i \to i+n_3}$ denote the backward label obtained by extending Γ_{i+n_3} to vertex i. Observe that the paths associated with Γ_i and $\Gamma_{i \to i+n_3}$ include the same sequence of stores after i and $i+n_3$, respectively. Under the assumptions of the proposition, label Γ_i dominates $\Gamma_{i \to i+n_3}$, i.e., $\Gamma_i \leq \Gamma_{i \to i+n_3}$. Indeed,

$\Gamma_i^{cost} \leq \Gamma_{i \to i+n_3}^{cost}$ since $\bar{c}_{i,i+n_3} = p_i - \lambda_i \geq 0$ by assumption, and $\bar{c}_{i+n_3,j} = \bar{c}_{ij}$ in view of Equation (4.5),
$\Gamma_i^{load} \leq \Gamma_{i \to i+n_3}^{load}$ since $d'_{i2} - d_{i2} \geq 0$ by assumption,
$\Gamma_i^{time} = \Gamma_{i \to i+n_3}^{time}$,
$\Gamma_i^k \leq \Gamma_{i \to i+n_3}^k$ for all stores k. □

4.4 Implementation

In order to obtain an efficient branch-and-price algorithm, we have implemented a number of different techniques. These techniques are partly associated with the column generation steps and partly related to the branch-and-bound process. They are briefly discussed in this section.

4.4.1 Lower bounding and upper bounding

In the course of branch-and-price we need to solve an LP relaxation via column generation in each node. In the root node we solve the LP relaxation of Problem (4.6)-(4.12) to obtain an initial Global Lower Bound (GLB), which is progressively updated by setting it equal to the lowest LP relaxation value among all nodes which are not pruned yet. In addition to this GLB, we consider a Local Lower Bound (LLB) in each node. By a node's LLB we mean a lower bound on the LP relaxation value (and hence, on the IP value) in that node. The LLB in each node is initially set to the optimal value of its parent node's LP relaxation. We exploit the LLBs in the following way. When the LP relaxation is solved in a specific node through column generation, it may happen that, due to degeneracy, new routes with negative reduced costs can still be found even though the optimal value of the LP has been reached. The LLBs help us to avoid such degenerate iterations in the nodes where the optimal value of the LP relaxation is equal to the initial LLB. It turns out that such nodes are abundant.

We also need a Global Upper Bound (GUB) during the branching procedure in order to fathom any node in which the associated LP relaxation value exceeds the GUB. Beside the classic way to improve the GUB (when a better integer solution is found at the end of column generation in any node), we have tested two additional ideas. First, we consider the formulation of the master problem obtained at the end of column generation at the root node, and we solve this formulation as an ILP problem, using CPLEX. Since this ILP problem only contains a restricted subset of routes, its optimal solution provides a heuristic solution (and hence, an initial upper bound) for the complete Formulation (4.6)-(4.12). This is computationally time consuming, but provides a feasible solution to the ILP immediately at the root node. Hence, it may yield a tight upper bound and decrease the total computation time by fathoming more nodes. Second, we record the objective function value for any integer solution we may find during the course of column generation, and we use it to improve the current GUB when possible. Note that even though the optimal solution of the LP relaxation in a node might not be integer, we may encounter some integer solutions during the course of column generation which are better than the GUB at hand. Investigating every solution to check whether it is integer takes some time, but it may also improve the GUB. We will describe the results of these tests in Section 4.5.

4.4.2 Branching

As suggested in the literature [Gutierrez-Jarpa et al., 2010], we branch on arcs even though decision variables in the master problem are routes. Indeed, if we branched on routes, fixing a route variable to zero would complicate the solution process. As proposed by many authors, based on the values of route variables in the master problem the value of arc (i, j) is calculated as follows:

$$x_{ij}^{(1)} = \sum_{r \in R_1 : (i,j) \in r} u_{r1} \tag{4.18}$$

$$x_{ij}^{(2)} = \sum_{r \in R_2 : (i,j) \in r} u_{r2} \tag{4.19}$$

When the optimal solution of the LP relaxation in some node is not integral we calculate $x_{ij}^{(1)}$ and $x_{ij}^{(2)}$ using Equations (4.18) and (4.19). Then, among all values $x_{ij}^{(1)}$, $i, j \in V$, and $x_{ij}^{(2)}$, $i, j \in V$, we branch on the variable with value closest to 0.5.

When setting an arc (i,j) to 0 in period 1 or 2 we eliminate any route in this period that includes arc (i,j). However, setting an arc to 1 needs more work. Indeed, when we set arc (i,j) to 1, many other arcs can be set to 0 as a direct consequence. Depending on the class to which stores i and j belong we can specify the arcs to be set to 0. Let us succinctly discuss through an example the relation between the arcs we have already branched on and the routes we should keep in the master problem.

Example. Consider the two-period VRP depicted in Figure 4.2, where $3 \in V_{+0}$ and $9 \in V_{++}$. If somewhere in the branching tree we branch on arc $(3,9)$ in period 1 and set $x_{39}^{(1)} = 1$, then, we can conclude that:

- $x_{93}^{(1)} = 0$ as we cannot simultaneously use arcs $(3,9)$ and $(9,3)$ in the same period in a solution;
- $x_{3j}^{(1)} = 0,\ \forall j \in V^0 \backslash \{9\}$ as the outgoing flow in period 1 from store 3 must be towards store 9;
- $x_{i9}^{(1)} = 0,\ \forall i \in V^0 \backslash \{3\}$ as the ingoing flow in period 1 to store 9 must be from store 3;
- $x_{3j}^{(2)} = 0,\ \forall j \in V^0$ and $x_{i3}^{(2)} = 0,\ \forall i \in V^0$ as store $3 \in V_{+0}$ and when it is served in period 1, it cannot be served in period 2;
- $x_{9,10}^{(2)} = 0$ and $x_{10,j}^{(2)} = 0,\ \forall j \in V^0$ as store $9 \in V_{++}$ and when it is served in period 1, the corresponding virtual store 10 cannot be served in period 2 (no postponing). $\square$

We have tested breadth-first, depth-first, and lowest-lower-bound-first strategies to explore the nodes of the branching tree. As the initial ILP solution has a very low optimality gap, improving the lower bound rather than the upper bound proves to have the best efficiency in our test instances. Our computational results are accordingly based on the lowest-lower-bound-first strategy.

4.4.3 Route generation

In each node, a set of initial routes should be introduced in the master problem to guarantee feasibility of the corresponding LP relaxation problem. We always include in this initial set all compatible routes of the father node. Then, new routes are introduced in the master problem by the column generation procedure until we reach optimality of the LP relaxation. These new routes must again be compatible with the status of the node in terms of the earlier branching decisions, where the status of a node consists of a set of arcs with value fixed to 1 and a set of arcs with value fixed to 0 for each period.

Compatibility in a given period means that neither the initial routes nor those generated during the course of column generation can include any of the arcs whose value is fixed to 0 in that period. Moreover, every arc whose value if fixed to 1 must be part of at least one of the routes in that period, in every iteration of column generation.

4.4.4 Column management

Classically, in each iteration of column generation we should independently solve a pricing problem for period 1 and a pricing problem for period 2. In each pricing problem the label-setting algorithm is used to find a route or a set of routes with negative reduced cost(s). In each period, we may stop expanding the routes in the algorithm once we obtain a route with a negative reduced cost. Alternatively, we could stop when we obtain a pre-specified number of routes with negative reduced costs or we could capture all of them. In our experiments, the column generation algorithm performs best when we simultaneously introduce around 100 routes with negative reduced costs in each iteration. Besides adding new routes to the master problem, some authors consider deleting inefficient routes, i.e., routes that do not appear in the optimal solution of the master problem during some consecutive iterations, or that have a very big positive reduced cost [Dell'Amico et al., 2006]. This did not prove useful in our experiments. So, we do not delete any route except for the sake of guaranteeing compatibility and feasibility.

4.4.5 Stabilization

Degeneracy is a very common phenomenon when we apply column generation [Lübbecke and Desrosiers, 2005]. Stabilization techniques can be used to decrease the number of degenerate solutions. To our knowledge, no absolute superiority of any stabilization technique over others has been reported yet [Lübbecke and Desrosiers, 2005]. We have implemented the stabilization technique introduced by Du Merle et al. [1999]. Our test results demonstrate that the efficiency of this technique highly depends on the number of routes introduced in the master problem in each iteration of column generation (the "bunch size"). When only a small number of routes are introduced in the master problem in each iteration, the stabilization technique helps in terms of decreasing the number of degenerate solutions, and hence the total computation time. On the other hand, if we introduce many routes in the master problem in each iteration of column generation, we are more likely to capture a new solution and so to avoid degeneracy. Following our discussion of column management in the previous subsection, we add up to 100 routes with negative reduced costs in the master problem in each period, and stabilization did not prove useful in this framework.

4.5 Computational results

The branch-and-price algorithm was coded in Java and the instances were run on an Intel i7 processor with 1.8GHz CPU and 8GB RAM. We used ILOG CPLEX 12.4 to solve the restricted master problems. A time limit of 600 seconds was set for each instance.

4.5.1 Instances

The 100-series instances created by Solomon [1987] were considered for the test problems. These include clustered stores (C101-C109), randomly distributed stores (R101-R112), and randomly-clustered stores (RC101-RC108). As in the original instances, time windows and service times are considered, the capacity of each truck is $Q = 200$, and the cost c_{ij} of traveling from store i to store j is equal to the Euclidean distance from i to j. Demands for the two periods are set as follows. Denote by d_i the demand of store i in a Solomon instance. In the two-period VRP, we let $d_{i1} = d'_{i2} = d_i$ for stores in V_{+0}, $d'_{i1} = d_{i2} = d_i$ for stores in V_{0+}, and $d_{i1} = d_{i2} = d_i$, $d'_{i1} = d'_{i2} = 2d_i$ for stores in V_{++}. The advancement penalty per unit is equal to 0.2 (respectively, 2 and 1) for class C (respectively, R and RC) instances. The postponement penalty is equal to twice the advancement penalty in all cases. This implies that for clustered instances, for example, $a_i = 0.2d_i$ and $p_i = 0.4d_i$. (These values have been adjusted in such a way that only 10-20% of orders are eventually shifted in all instances C, R, and RC.)

We consider 29 medium-size and 29 large instances. The medium-size instances consist of 50 stores; they include 20 stores in each of the classes V_{0+} and V_{0+}, and 10 stores in class V_{++}. This implies that when dealing with the two-period VRP, we are essentially dealing with two interdependent VRPs, each involving 50 real stores and 10 virtual stores. Our large instances contain 70 stores, with 25, 25, and 20 stores in each class, respectively. So, when shifts are allowed, the two-period VRP consists of two interdependent VRPs, each with 70 real stores and 20 virtual ones.

Since all Solomon instances of each class (C, R, and RC) consider the same locations for all stores (e.g., the coordinates of store 1 are the same in all instances R101-R112), we further diversified our instances, as follows. Each of our instances can be described by the notation Aα-β, where A denotes one of the Solomon classes C, R, or RC; α denotes an instance identifier in {101,...,112}; and β is a store identifier in {01,...,91}. The instance Aα-β involves the set of stores $\{\beta, \ldots, \beta + N\}$ from the Solomon instance Aα, where $N = n_1 + n_2 + n_3$ is either 50 or 70, depending on the instance size. Thus, for example, our medium-size instance R102-21 contains stores 21-70 from Solomon instance R102, with stores 21-40 in V_{+0}, stores 41-60 in V_{0+}, and stores 61-70 in V_{++}. The complete list of all instances that we have considered can be read from Tables 4.6-4.7.

4.5.2 Presentation of results

The numerical results are presented in Tables 4.6-4.7. The column header A0P0 refers to the solution of the problem where neither advancing nor postponing is allowed (or equivalently, the solution of two independent VRP problems), and the header A1P1 refers to the two-period VRP problem, where both advancing and postponing are allowed. The column labeled "Gap" shows a bound on the optimality gap of the best integer solution we find in the branching tree within the time limit; more precisely, Gap = (UB - LB)/UB, where UB and LB are, respectively, the best available upper bound and lower bound on the total cost. The next columns, respectively labeled "Z", "#Veh.", and "Time", display the total cost over both periods, the total number of vehicles used in both periods, and the total solution time (in seconds). A dash sign (-) indicates instances for which we could not even solve the LP relaxation in the root node within the time limit. Column "%Z Imp." displays the percentage improvement of the total cost obtained for A1P1 with respect to the best cost obtained for A0P0. Finally, "#Adv." and "#Pos." indicate the number of advancements and postponements in the best solution of A1P1, respectively.

4.5.3 Difficulty of the instances

Tables 4.6-4.7 show that, within 600 seconds, our algorithm has solved 46 out of 58 instances of A0P0 to optimality, and 56 out of 58 instances of A0P0 within 5% of optimality. The total number of instances of A1P1 solved to optimality within the same time limit is 33, and 53 instances of A1P1 have been solved within 7% of optimality. Restricting our attention to those 33 instances which are solved to optimality in both A0P0 and A1P1, we see that when delivery shifts are allowed (A1P1), the average

Table 4.6: Medium-size instances 20-20-10

Instance	A0P0				A1P1					
	Gap	Z	#Veh.	Time	Gap	%Z Imp.	#Veh.	#Adv.	#Pos.	Time
C										
101-11	0	594.9	7	4	0	6.1	7	0	2	5
102-21	0	630.8	7	2	0	18.4	6	4	3	48
103-31	0	684.6	7	198	-	-	-	-	-	600
104-41	-	-	-	600	-	-	-	-	-	600
105-51	0	750	7	2	0	11.9	7	3	2	371
106-61	0	556.9	7	75	0	14.5	6	2	0	233
107-71	0	679.9	9	1	0	8.1	7	2	4	33
108-81	0	510.4	6	63	.06	0	6	0	0	600
109-91	.03	621.6	8	600	.04	10.8	7	4	1	600
R										
101-11	0	1373.1	19	0	0	5.9	17	6	0	0
102-21	0	1579.1	19	0	0	5.7	16	10	0	0
103-31	0	1119.3	13	1	0	3.7	12	3	1	15
104-41	0	912	9	140	0	4	8	3	0	272
105-51	0	1027.1	12	1	0	4.3	11	2	1	3
106-61	0	939.6	11	2	0	1.1	11	2	0	22
107-71	0	827.7	9	61	0	3.7	8	3	0	115
108-81	0	764.5	7	206	.02	0	7	0	0	600
109-91	0	1048.1	11	23	0	4	10	3	0	8
110-01	0	1009.8	10	14	0	1.8	10	0	2	281
111-11	0	888.5	10	4	0	1.7	10	3	0	33
112-21	0	976.4	9	102	0	1.7	8	2	1	580
RC										
101-11	0	1539.6	15	131	0	17.3	10	12	1	61
102-21	0	1232	13	7	.01	6.3	10	4	2	600
103-31	0	1225.4	11	280	0	12	9	5	2	251
104-41	0	976	8	64	.01	3.2	7	4	2	600
105-51	0	1048	11	0	0	3.4	11	4	2	11
106-61	0	1026.6	11	14	0	3	10	6	0	97
107-71	.01	1037.8	12	600	.02	2.7	11	1	2	600
108-81	0	893.6	9	16	0	5.1	8	5	1	410

computation time to solve an instance to optimality increases by a factor of 20 as compared to solving two independent VRPs (A0P0). Thus, the two-period VRP appears to be considerably harder than the classical VRP.

On the other hand, comparing the number of clustered instances (C) solved to optimality with the number of random instances (R) solved to optimality demonstrates that the former ones are more difficult in the two-period VRP context. This is contrary to what we usually observe for the Solomon instances in the classical VRP case. A closer look reveals that, when only one period is considered, the structure of the VRP optimal solution in the vast majority of clustered instances is such that each cluster is served by only one vehicle, and each vehicle serves only one cluster. This very simple structure results in small enumeration trees and in low computation times. However, the solutions of the two-period VRP do not display the same structure. In our C-instances of A1P1, it frequently happens that some vehicle serves stores in two or even three clusters, and some clusters are served by more than one vehicle. As a consequence, even solving the LP relaxations of the clustered instances is very difficult. Besides, integer solutions are rarely obtained at the root node as the result of solving the restricted master problem, unlike the VRP case where the clustered instances are frequently solved

Table 4.7: Large instances 25-25-20

	A0P0				A1P1					
Instance	Gap	Z	#Veh.	Time	Gap	%Z Imp.	#Veh.	#Adv.	#Pos.	Time
C										
101-11	0	924.1	11	2	.07	8.3	10	8	0	600
102-21	0	880.6	11	7	.07	5.6	9	9	8	600
103-31	0	934	10	310	-	-	-	-	-	600
104-41	-	-	-	600	-	-	-	-	-	600
105-51	0	954.7	11	130	.04	11	10	10	1	600
106-61	0	814.4	11	4	0	9.8	9	8	2	390
107-71	0	930.5	12	5	0	9.9	10	8	1	375
108-81	0	792.5	10	84	.06	0.3	10	11	0	600
109-91	.01	808	10	600	-	-	-	-	-	600
R										
101-11	0	1986	24	0	0	4.9	23	3	1	1
102-21	0	1809.2	25	1	0	8.3	20	6	1	1
103-31	0	1302.3	15	26	0	2.9	14	5	0	41
104-41	.01	1089.2	13	600	.01	2	12	4	1	600
105-51	0	1387.6	18	125	.01	2.2	16	7	0	600
106-61	0	1347.7	15	7	0	2.2	14	5	0	30
107-71	0	1210.2	13	51	0	4.5	11	6	1	309
108-81	.03	1115.4	11	600	.02	3.3	10	3	1	600
109-91	0	1261.3	15	7	0	2.5	14	5	2	377
110-01	0	1313.7	13	276	0	2.1	12	5	0	321
111-11	0	1304.8	13	272	0	3.5	12	6	0	200
112-21	.01	1118.3	11	600	.04	0.1	11	1	0	600
RC										
101-11	0	1968.1	20	22	0	11.7	15	12	3	574
102-21	.01	1796.5	17	600	.04	7.4	15	13	1	600
103-31	.02	1331.4	12	600	.03	3.5	11	3	3	600
104-41	.03	1173.2	11	600	.02	2	10	4	0	600
105-51	0	1607.4	17	4	0	5.4	15	5	1	58
106-61	0	1460	13	71	.02	0	13	7	0	600
107-71	0	1375.5	14	592	.03	0	14	0	0	600
108-81	.05	1222	12	600	.01	5.9	11	1	0	600

to optimality at the root node (see also Section 4.5.4). So, branching is necessary in most instances of A1P1.

4.5.4 Algorithmic insights

As explained in Section 4.4.1, CPLEX is used to solve the ILP defined by the current formulation of the master problem at the root node (with the limited number of routes obtained at the end of column generation). This process already delivers an integer feasible solution of the two-period VRP at the root node, before any branching. Figure 4.4 shows the gap between the value of this feasible solution and the optimal value of the two-period VRP for all 33 instances of A1P1 solved to optimality within the time limit. The average optimality gap of the ILP solution at the root node is only 0.6% for these instances. Figure 4.5 compares the computation times required, respectively, to obtain the ILP solution at the root node and to complete the branch-and-price procedure for the same instances. The average computation time for the ILP solution is 25% of the time required to perform the branch-and-price process to optimality. (We obtain similar figures if we extend the analysis to include all A0P0 instances solved to optimality.)

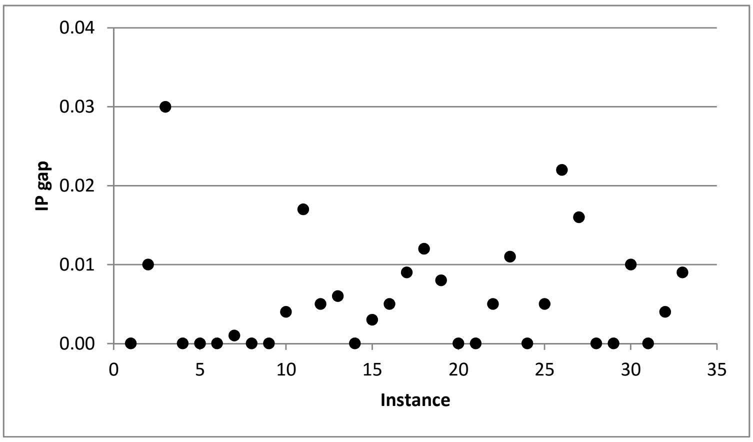


Figure 4.4: ILP gap at the root node for the instances solved to optimality.

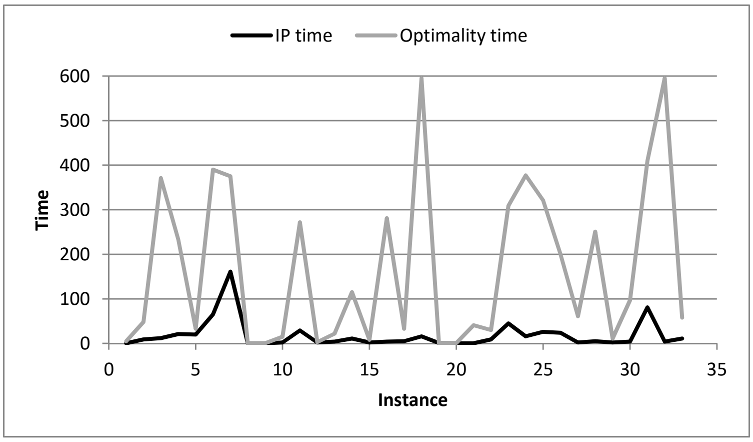


Figure 4.5: ILP time at root node and total solving time for the instances solved to optimality.

So, Figures 4.4-4.5 show that by spending less than one fourth of the time required to complete the branch-and-price process, we can obtain a very good solution with optimality gap smaller than 1%. Besides, the availability of this good initial solution also suggests that an efficient strategy to bridge the optimality gap is to improve the global lower bound. In other words, exploring the nodes with the lowest lower bounds first is likely to be the best node exploration strategy, and this is indeed confirmed by our experiments.

In contrast with the previous observations, solving an ILP formulation in every node of the branch-and-price tree does not significantly improve the upper bound and does not reduce the total solution time. Similarly, the second algorithmic idea mentioned in Section 4.4.1 (i.e., to record the integer solutions obtained by chance during the course of column generation) did not prove very useful, because such integer solutions rarely occur in practice.

As expected from the VRP literature, all three acceleration techniques mentioned in Section 4.3.5 significantly enhance the efficiency of the label-setting procedure. More interestingly, perhaps, the results presented in Section 4.3.6 also prove quite useful to solve the two-period VRP. Indeed, over all medium and large instances of A1P1 (except C103, C104 and C109-large, whose LP relaxations could not be solved within the time limit), the computation time to solve the LP relaxation in the root node decreases, on average, by 34% when using Proposition 1 and Proposition 2, as opposed to not using them. The gain is of 3% for C-instances, 49% for R-instances, and 38% for RC-instances. (Recall that the shifting penalty per unit is equal to 0.2, 2, and 1, respectively, for instances of type C, R, and RC, and observe that the propositions are more likely to be effective when the penalties get larger.)

Finally, regarding the branching process, let us simply mention that considering all compatible routes of a father node when we start solving the LP relaxation in a child node, as explained in Section 4.4.3, greatly increases the speed of our branch-and-price algorithm.

4.5.5 Managerial insights

Here, we analyze the main benefits of allowing deliveries to be shifted. We also briefly discuss how we can identify the deliveries which are very unlikely to be shifted in an optimal solution.

4.5.5.1 Benefits of allowing shifts.

By allowing deliveries to be shifted, the two-period VRP model logically yields lower total transportation costs over two periods than two independent VRP models. Comparing model A1P1 against the basic model A0P0 provides an estimate of the improvement. Tables 4.6-4.7 indicate that the average cost reduction over all instances is 5.3%; it is 6.2% over all instances of A1P1 solved to optimality. In both cases, the gain is economically significant.

More generally, Table 4.8 synthesizes the results obtained when positive shifting penalties are applied as in A1P1, whereas Table 4.9 shows the results when shifts are allowed without any penalty, i.e., $p_i = a_i = 0$ for all stores. The latter case indicates the maximum savings we may reap in the two-period model, under the most favorable circumstances. Table 4.8 contains results for the 33 instances of A1P1 solved to optimality. When no penalties are applied, 18 instances of A1P1 are solved to optimality based on which the results are reported in Table 4.9. The value "%Shift" shows the average percentage of the number of orders advanced or postponed, out of the number of orders which could be shifted, namely, $n_1 + n_2 + n_3$. The value "%Z Imp." shows the average percentage of reduction in transportation costs in both periods, with respect to A0P0. The values "%Occ." and "%Veh." show the average percentage of improvement in the occupation of the vehicles and in the number of vehicles used, respectively, compared to the basic model A0P0.

We observe that the average cost improvement when penalties are zero is 26.6%. Furthermore, the total number of vehicles decreases by 26.2% on average under the same assumption and, as a direct consequence, the vehicle occupation rate increases by 38.6%. Of course, such spectacular improvements may not be feasible in practice, since the negative impact of shifting deliveries cannot be totally disregarded by the stores. But these figures provide an indirect estimate of the cost of *not* allowing any

Table 4.8: Average improvements with respect to model A0P0 for different penalties (instances of A1P1 solved to optimality)

Instances	Penalty	%Shift	%Z Imp.	%Occ.	%Veh.
C	0.2	10.2	11.2	14.7	-12.2
R	2	7.9	3.6	9.6	-8.4
RC	1	15.1	8.3	20	-15.5
Weighted average		9.9	6.2	12.9	-10.7

Table 4.9: Average improvements with respect to model A0P0 for different penalties (instances of A1P1 solved to optimality)

Instances	%Shift	%Z Imp.	%Occ.	%Veh.
C	49.6	18.3	12.1	-10.2
R	62.3	30.3	49.4	-32.5
RC	60.1	27.0	45.3	-31.1
Weighted average	58.5	26.6	38.6	-26.2

shifts; and they may be used by the managers of the system to evaluate the potential benefits that can be reaped from more flexible inventory control policies.

4.5.5.2 Identifying unpromising shifts.

It may also be interesting for managers to understand the circumstances under which it is profitable to shift deliveries from one period to another one. More precisely, we are going to show that it may be possible to identify *ex ante unpromising shifts*, that is, deliveries which are very unlikely to be shifted in the optimal solution.

Intuitively, shifting the delivery to store i is unpromising if the shifting penalty is relatively large with respect to the potential saving in routing costs that can be achieved by the shift. The value (p_i or a_i) of the penalty is known exactly, but the value of the saving depends on routing decisions and can only be roughly estimated. For this purpose, let us introduce the following notations. For a store $i \in V_{+0} \cup V_{++}$, we let $J_1(i) \subseteq V_{+0} \cup V_{++}$ be a subset of stores which are *accessible neighbors* of i, in the sense that for any store $j \in J_1(i)$, the distance from i to j is small (in our experiments, we include the five nearest stores to i), and the time windows do not prevent deliveries to both i and j on the same route in period 1. We similarly define $J_2(i) \subseteq V_{0+} \cup V_{++}$ for a store $i \in V_{0+} \cup V_{++}$.

We denote by $\tilde{c}_{iJ_1(i)}$ (respectively, $\tilde{c}_{iJ_2(i)}$) the average distance between store i and its accessible neighbors in period 1 (respectively, period 2). Then, we estimate the saving in routing costs resulting from postponing the delivery to store i as $\tilde{c}_{iJ_1(i)} - \tilde{c}_{iJ_2(i)}$ when $i \in V_{+0}$, and $\tilde{c}_{iJ_1(i)}$ when $i \in V_{++}$. The saving obtained by advancing a delivery from period 2 to period 1 is similarly estimated as $\tilde{c}_{iJ_2(i)} - \tilde{c}_{iJ_1(i)}$ when $i \in V_{0+}$, and $\tilde{c}_{iJ_2(i)}$ when $i \in V_{++}$.

Now, we regard a postponement of the delivery to store $i \in V_{+0} \cup V_{++}$ as unpromising if conditions (4.20) hold, and an advancement of the delivery to store $i \in V_{0+} \cup V_{++}$ as unpromising if conditions (4.21) hold.

$$\begin{cases} p_i > \tilde{c}_{iJ_1(i)} - \tilde{c}_{iJ_2(i)} & \forall i \in V_{+0} \\ p_i > \tilde{c}_{iJ_1(i)} & \forall i \in V_{++} \end{cases} \tag{4.20}$$

$$\begin{cases} a_i > \tilde{c}_{iJ_2(i)} - \tilde{c}_{iJ_1(i)} & \forall i \in V_{0+} \\ a_i > \tilde{c}_{iJ_2(i)} & \forall i \in V_{++} \end{cases} \tag{4.21}$$

The accuracy of the above criteria can be measured by the percentage of unpromising stores whose demand is actually shifted in the optimal solution of any instance. This percentage, say α, can be interpreted similarly to a type I error in statistics, since identifying a shift as "unpromising" may lead the manager to reject shifts that are potentially profitable. In our computational experiments (on 33 instances of A1P1 solved to optimality), $\alpha = 4\%$, which suggests that the impact of forbidding shifts of deliveries to unpromising stores should be limited. In fact, a more precise estimate of this economic impact can be obtained as follows: when we forbid shifts to unpromising stores and solve the resulting model to optimality, the total cost increases, on average, by only 2.3% over the same 33 instances.

These observations may also prove useful in a computational framework. Indeed, we have already observed that the two-period VRP tends to be much harder than a classical VRP. Therefore, it may be worth devising heuristic approaches for the solution of very large scale instances. If we use conditions (4.20)-(4.21) to identify unpromising shifts and we accordingly exclude the corresponding stores from consideration in period 1 or 2, the number of nodes of the networks decreases by 45% on average. Such a huge decrease of the network size might be very helpful when dealing with large instances in a heuristic fashion.

4.5.5.3 Proposing shifting penalties to the stores.

When a zero penalty is considered, the maximum saving in routing costs is attained. The saving is the difference between the sum of the optimal objective values of two independent VRPs and the optimal objective value of the two-period VRP. This saving can be regarded as the maximum penalty that can be paid to the whole network by the LSP. However, a rational way is that both sides (the LSP and the retail chain) share this saving. A caveat is that when a zero penalty is considered to solve the two-period VRP, there are often multiple optimal solutions. It makes sense to pick up the optimal solution with the least number of delivery shifts. To this end, we can consider a very small shifting penalty for each store when solving the two-period VRP.

When stores' benefits are independent of each other, it is up to the LSP proposing a shifting penalty to each store. One simple way is to share (part of) the saving with the stores by allocating to each store a penalty proportionate to the quantity of its shifted delivery. Note that this strategy works if all stores with shifted deliveries in the optimal solution of the two-period VRP accept the shift. Otherwise, even if one store does not accept the shift proposed by the optimal solution, the whole saving may ruin.

When it is difficult or unlikely to get an affirmative response from all stores with shifted deliveries, we may follow a sequential strategy which may often result in a smaller saving value. To this end, first we solve two independent VRPs in periods 1 and 2 without any shift. Let Z_0 denote the sum of the objective values of these base VRP models in periods 1 and 2. Let Z_j denote the sum of the objective values of the VRP models where the delivery to only store j is fully shifted to period 1. The value of $(Z_0 - Z_j)^+$ shows the maximum penalty that can be paid to store j for advancing its delivery. Similarly, advancing and postponing penalties for all stores can be calculated. Then, the LSP may propose (a percentage of) the biggest penalty to the associated store. If the store accepts the penalty, its initial delivery period is shifted and fixed; otherwise, the store with the next biggest penalty is chosen. This procedure continues by solving the new VRP models in periods 1 and 2, where the stores which have already accepted the penalties and so the shift proposals are repositioned to receive service in the corresponding period. The search stops when all non-fixed stores reject the shifting proposals. Obviously, the total saving that the LSP may gain in this strategy is less than the total attainable saving through solving the two-period VRP with zero penalties, as the former strategy is a greedy local search.

4.6 Conclusions

We have introduced a two-period VRP where orders of each period can be shifted to the other period and change in quantity. An efficient branch-and-price algorithm based on classical techniques from the VRP literature (column generation, label-setting algorithm, branching process, etc.) has been implemented to solve this model. Two new acceleration techniques that exploit the specific features of our model have also been developed; they significantly increase the efficiency of the label-setting algorithm. We have investigated the quality of the solution obtained by solving an ILP at the end of the column generation phase at the root node. Our experimental results show that this heuristic solution provides a very tight initial upper bound. As a consequence, and even though the two-period VRP turns out to be considerably harder than the classical VRP, our algorithms yield provably good solutions for many instances of the problem. In terms of managerial impact, the experiments demonstrate that the routing costs and the number of vehicles can decrease significantly when orders are allowed to be shifted. In other words, there is potential value in handling the 2-VRP model, as opposed to solving two independent VRP models. The results also suggest that, if one wants to avoid the computational burden of solving large two-period VRPs to optimality, identifying unpromising shifts may reduce the size of the instances to be solved while still producing economies in transportation costs.

Chapter 5

A two-period vehicle routing problem with partial delivery shifts

5.1 Introduction

Consider the two-period VRP discussed in Chapter 4. Assume that the sum of delivery quantities in two periods requested by each store is a fixed quantity. While delivering such a quantity in two periods is a hard constraint to be respected, the LSP is free to deliver any quantity in each period. However, any diversion from the initial orders placed by each store is penalized. In other words, the LSP may decide to postpone part of a delivery requested for period 1 and to deliver it in period 2 along with the initial delivery requested for period 2. In this case, the LSP is charged a financial penalty per unit. Similarly, advancing part of the initial delivery requested for period 2 to period 1 could be acceptable for a store, but the LSP has to pay a penalty for it. We assume linear penalties for shifts based on the quantity shifted, though advancements and postponements may impose different penalties. If advancement or postponement is not allowed by a store, the associated penalties may be considered as so high that they effectively deter the LSP from performing the shifts. The LSP's objective is to minimize the sum of the routing costs in two periods and of the penalties for the shifted deliveries.

In this chapter, we present two as an mixed integer linear programming (MILP) formulations of the two-period VRP with partial delivery shifts. We describe a column-and-row generation algorithm to solve the LP-relaxation of the first MILP formulation. A difficulty with this approach is that, whenever a new column is introduced in the model, a new capacity constraint must be taken into account as well. We explain how the column-generation procedure can be tailored to circumvent this difficulty. We also develop a column generation algorithm to solve the LP-relaxation of the second MILP. Computational experiments will be required, in future work, to determine the performance of each approach.

5.1.1 Motivation

Consider the two-period VRP with full delivery shifts discussed in Chapter 4. Provided that the sum of the delivery quantities in two periods is a fix quantity and the delivery quantities are continuous, the two-period VRP with partial delivery shifts is a more general model than the one with full delivery shifts. Therefore, the optimal solution of the two-period VRP with partial delivery shifts provides a better value for the objective function, consisting of routing and penalty costs, as compared to the two-period VRP with full delivery shifts.

5.1.2 Scientific contributions

The main contributions of this chapter can be summarized as follows.

- We formulate the two-period VRP as a MILP problem in two different ways, and we establish a one to one relation between feasible solutions of the two formulations.
- We develop a column-row generation algorithm to solve the LP-relaxation of the first MILP formulation and a column generation algorithm to solve the LP-relaxation of the second MILP formulation.
- Label-setting algorithms are described to solve the pricing problem raised in each formulation.

This chapter is organized as follows. A full description of the problem, including the required notations, is provided in Section 5.2. The first MILP formulation is presented in Section 5.3, where we also analyze the master problem and its dual problem to draw a criterion on detecting new promising routes to be introduced to the master problem. Section 5.4 presents the second MILP formulation and a column generation algorithm to solve its LP-relaxation. Finally, concluding remarks are provided in Section 5.6.

5.2 Problem statement

In the two-period VRP defined in Section 5.1, assume that: (1) the delivery quantities are continuous, (2) the sum of the delivery quantities in two periods is a fixed value for each store, and (3) advancing and postponing penalties are per proportional to the quantity shifted. We use the notations in Tables 5.1-5.3. For the sake of convenience, we will redefine some of them in the course of our discussion.

Define d_{i1} and d_{i2} as the orders of store i for periods 1 and 2, respectively. Each vehicle can perform at most one single route per period. Split deliveries within a period are not allowed, that is, each store is served by at most one vehicle in each period. We distinguish three classes of stores, i.e., V_{+0}, V_{0+}, and V_{++} as defined in Chapter 4. For stores in class V_{+0}, if the LSP decides to shift part of the order to period 2, it is charged a penalty p_i for each unit of the postponed delivery. Similarly, for stores in class V_{0+}, the LSP can deliver a positive quantity in period 1 , but is charged a penalty a_i for each unit of the advanced delivery. Finally, for stores in class V_{++}, either part of the initial order d_{i1} can be postponed or part of the initial order d_{i2} can be advanced. For each case, per unit penalties p_i and a_i are incurred for postponing and advancing, respectively. If either p_i or a_i is infinite, we say that the corresponding shift is *forbidden*.

Table 5.1: Indices and sets

i, j	indices for vertices (stores)
r	index for routes
V_{+0}	set of stores with a positive order for period 1 and no order for period 2
V_{0+}	set of stores with no order for period 1 and a positive order for period 2
V_{++}	set of stores with positive orders for both periods
V	$V_{+0} \cup V_{0+} \cup V_{++}$
V^0	$V \cup \{0\}$ where vertex 0 denotes the depot
A	set of arcs
$A(r)$	set of arcs in route r
R_t	set of feasible routes in period t

Table 5.2: Parameters

d_{it}	order of store i for period t $t = 1,2$
d_i	order of store i for two periods $(d_i = d_{i1} + d_{i2})$
p_i	per unit postponement penalty imposed by store i
a_i	per unit advancement penalty imposed by store i
n_1	number of stores in set V_{+0}
n_2	number of stores in set V_{0+}
n_3	number of stores in set V_{++}
n	total number of stores $(n = n_1 + n_2 + 2n_3)$
c_{ij}	cost of using arc (i, j)
Q	capacity of each vehicle
t_{ij}	travel time to traverse arc (i, j)
s_i	service time at store i
(e_i, l_i)	time window for the arrival of a vehicle at vertex i
α_{ir}	1 if store i belongs to route r; 0 otherwise.

5.3 Column-row generation

In this section, we present a formulation of the two-period VRP with partial delivery shifts. We will also develop a column-row generation algorithm to solve its LP-relaxation, where the pricing problem is an ESPPRC.

5.3.1 A mixed integer linear programming formulation

A MILP formulation of the two-period VRP can be obtained, as for the VRP, by introducing decision variables corresponding to the selection of feasible routes in each period; see Table 5.3, where R_t denotes the set of feasible routes in period t. A route is feasible if (1) it starts and ends at the depot and visits each vertex at most once (elementarity) and (2) if a store is visited, the visit is within its time window. Contrary to the usual definition of a feasible route in VRP context, respecting vehicle capacity is not considered in the definition of a feasible route, but rather it will be included in our ILP formulation as a constraint; see problem formulation in Section 5.3.1.

Table 5.3: Decision variables

u_{rt}	if route $r \in R_t$ is used in period t; 0 otherwise
y_{irt}	delivery quantity to store i by route r in period t
w_{it}	excess delivery quantity to store $i \in V_{++}$ in period t

The two-period VRP with partial delivery shifts can be formulated as the following MILP problem, where $c_r = \sum_{(i,j)\in A(r)} c_{ij}$.

$$\begin{aligned} \min & \sum_{r\in R_1} c_r u_{r1} + \sum_{r\in R_2} c_r u_{r2} \\ + & \sum_{r\in R_2} \sum_{i\in(r\cap V_{+0})} p_i y_{ir2} + \sum_{r\in R_1} \sum_{i\in(r\cap V_{0+})} a_i y_{ir1} \\ + & \sum_{i\in V_{++}} (a_i w_{i1} + p_i w_{i2}) \end{aligned} \tag{5.1}$$

subject to

$$\sum_{r\in R_1} \alpha_{ir} u_{r1} \leq 1;\ \forall i \in V \tag{5.2}$$

$$\sum_{r\in R_2} \alpha_{ir} u_{r2} \leq 1;\ \forall i \in V \tag{5.3}$$

$$\sum_{r\in R_1} y_{ir1} + \sum_{r\in R_2} y_{ir2} = d_i;\ \forall i \in V \tag{5.4}$$

$$\sum_{r\in R_1} y_{ir1} - w_{i1} \leq d_{i1};\ \forall i \in V_{++} \tag{5.5}$$

$$\sum_{r\in R_2} y_{ir2} - w_{i2} \leq d_{i2};\ \forall i \in V_{++} \tag{5.6}$$

$$\sum_{i\in r} y_{ir1} - Q u_{r1} \leq 0;\ \forall r \in R_1 \tag{5.7}$$

$$\sum_{i\in r} y_{ir2} - Q u_{r2} \leq 0;\ \forall r \in R_2 \tag{5.8}$$

$$u_{rt} \in \{0,1\};\ \forall r \in R_t,\ t = 1,2 \tag{5.9}$$

$$y_{irt} \geq 0;\ \forall i \in V,\ r \in R_t,\ t = 1,2 \tag{5.10}$$

$$w_{it} \geq 0;\ \forall i \in V_{++},\ t = 1,2 \tag{5.11}$$

The objective function (5.1) consists of fixed and variable costs of each route in both periods. It also encompasses postponement penalty for any store in class V_{+0} if it is included in a route selected in period 2, advancement penalty for any store in class V_{0+} if it is included in a route selected in period 1, postponement penalty for any store in class V_{++} if extra quantity is delivered to the store in period 2, and advancement penalty for any store in class V_{++} if extra quantity is delivered to the store in period 1. Constraints (5.2) and (5.3) guarantee that every store is served at most once in each period. Constraints (5.4) guarantee that a total required demand for two periods is actually delivered to each store. Before explaining the next constraints, note that $w_{it} = (\sum_{r\in R_t} y_{irt} - d_{it})^+$ shows the excess delivery quantity to store i in period t. This nonlinear equality can be expressed by the linear constraints (5.5), (5.6), and (5.11). This is due to the positive coefficient of w_{it} in the objective function of the minimization problem. Constraints (5.7) and (5.8) guarantee feasibility of each route in terms of vehicle capacity in periods 1 and 2, respectively. Finally, constraints (5.9)–(5.11) impose integrality and nonnegativity of the decision variables. Note that constraints (5.4) can be replaced by constraints (5.12) without impacting the optimal solution.

$$\sum_{r\in R_1} y_{ir1} + \sum_{r\in R_2} y_{ir2} \geq d_i;\ \forall i \in V \tag{5.12}$$

This is due to the positive coefficient of variables y_{irt} in the objective function. Moreover, the following inequalities are also valid for Problem (5.1)–(5.11),

$$y_{irt} \leq d_i u_{rt};\ \forall i \in V,\ r \in R_t,\ t = 1,2 \tag{5.13}$$

$$\sum_{r\in R_1} \alpha_{ir} u_{r1} + \sum_{r\in R_2} \alpha_{ir} u_{r2} \geq 1;\ \forall i \in V \tag{5.14}$$

Problem (5.1)–(5.11) cannot be solved by a classical column generation algorithm since it contains a constraint for each route. Consider the LP-relaxation of Problem (5.1)-(5.11) where Constraints (5.4) are replaced by Constraints (5.12) and other constraints are transformed into the form of $\geq$ so that all the corresponding dual variables are non-negative. The resultant problem is called the master problem. Consider a particular route r and a particular period t. By column set (r,t), we mean columns of technological coefficients of decision variables u_{rt} and y_{irt} for every $i \in r$. If only a limited number of column sets and all variables w_{it} are taken into account in the master problem, the problem is called the restricted master problem.

The master problem defined contains too many column sets. One way to solve this is by considering a restricted master problem and solving it to optimality. Then, new promising column sets are identified and added to the restricted master problem and it is solved again. This procedure continues until no further improvement is possible, i.e., until no new column set is able to improve the objective value.

In the following two subsections, we look into the master problem from two different angles in order to identify new promising column sets. Nevertheless, we will see that both ways lead to the same criterion to identify new promising column sets.

5.3.2 Analysis of the primal master problem

Consider a Restricted Master Problem (RMP) solved to optimality by taking into account decision variables w_{it} for every $i \in V_{++}$ and $t = 1,2$, and a limited pool of column sets associated with routes in R_t for $t = 1,2$. From now on, with some small abuse of notations, by R_t we mean a subset of (and not all) feasible routes in period t in terms of elementarity and time windows. We may assume that indeed all column sets existed in the pool, but they have been completely neglected when optimizing the master problem. Now, we look at a generic column set in one of the periods, say column set $(r', t = 1)$.

When activating column set $(r', 1)$, variables $u_{r'1}$ and $y_{ir'1}$ for every $i \in r'$ should be regarded in the model. Moreover, a constraint imposing feasibility of route r' in terms of the vehicle capacity is activated and added to the constraints. The RMP containing these additional variables is written as follows.

$$\begin{aligned} \min & \sum_{r \in R_1} c_r u_{r1} + \sum_{r \in R_2} c_r u_{r2} \\ & + \sum_{r \in R_2} \sum_{i \in (r \cap V_{+0})} p_i y_{ir2} + \sum_{r \in R_1} \sum_{i \in (r \cap V_{0+})} a_i y_{ir1} \\ & + \sum_{i \in V_{++}} (a_i w_{i1} + p_i w_{i2}) \\ & + c_{r'} u_{r'1} + \sum_{i \in (r' \cap V_{0+})} a_i y_{ir'1} \end{aligned} \tag{5.15}$$

subject to

$$-\sum_{r \in R_1} \alpha_{ir} u_{r1} - \alpha_{ir'} u_{r'1} \geq -1; \ \forall i \in V, \ (\text{dual variables: } \beta_i) \tag{5.16}$$

$$-\sum_{r \in R_2} \alpha_{ir} u_{r2} \geq -1; \ \forall i \in V, \ (\text{dual variables: } \gamma_i) \tag{5.17}$$

$$\sum_{r \in R_1} y_{ir1} + \sum_{r \in R_2} y_{ir2} + y_{ir'1} \geq d_i; \ \forall i \in V, \ (\text{dual variables: } \delta_i) \tag{5.18}$$

$$-\sum_{r \in R_1} y_{ir1} + w_{i1} - y_{ir'1} \geq -d_{i1}; \ \forall i \in V_{++}, \ (\text{dual variables: } \rho_i) \tag{5.19}$$

$$-\sum_{r \in R_2} y_{ir2} + w_{i2} \geq -d_{i2}; \ \forall i \in V_{++}, \ (\text{dual variables: } \eta_i) \tag{5.20}$$

$$-\sum_{i \in r} y_{ir1} + Q u_{r1} \geq 0; \ \forall r \in R_1, \ (\text{dual variables: } \theta_r) \tag{5.21}$$

$$-\sum_{i \in r} y_{ir2} + Q u_{r2} \geq 0; \ \forall r \in R_2, \ (\text{dual variables: } \pi_r) \tag{5.22}$$

$$-\sum_{i \in r'} y_{ir'1} + Q u_{r'1} \geq 0; \ (\text{dual variable: } \theta_{r'}) \tag{5.23}$$

$$u_{rt}, \ y_{irt}, \ w_{it}, \ u_{r'1}, \ y_{ir'1} \geq 0 \tag{5.24}$$

The criterion to identify a new promising column set to add to the master problem is stated in Proposition 5. However, we need Propositions (3) and (4) in order to prove Proposition 5.

Proposition 3. *Consider a square invertible matrix B of size $m\times m$, and an extended matrix A of size $(m+1)\times(m+1)$:*

$$A=\begin{bmatrix} B_{m\times m} & h_{m\times 1}\\ 0_{1\times m} & q_{1\times 1}\end{bmatrix}$$

where, $q\neq 0$. The inverse of matrix A is calculated as follows:

$$A^{-1}=\begin{bmatrix} B^{-1} & -\frac{1}{q}B^{-1}h\\ 0 & \frac{1}{q}\end{bmatrix}$$

Proof. It suffices to check that:

$$A\cdot A^{-1}=\begin{bmatrix} I_{m\times m} & 0_{m\times 1}\\ 0_{1\times m} & 1_{1\times 1}\end{bmatrix}$$

□

In Proposition (4), we show how the value of the dual variable associated with a new constraint is calculated before actually adding the constraint to the master problem. Based on the result of this proposition, we will calculate the reduced cost of each variable in a generic column set (r,t).

Proposition 4. *Assume that column set $(r',1)$ is added to the master problem along with the associated vehicle capacity Constraint (5.23). The value of the dual variable of Constraint (5.23) in the current solution is calculated as $\theta_{r'}=\frac{1}{Q}(c_{r'}+\sum_{j\in r'}\beta_j)$.*

Proof. Let $\alpha_{r'}$ denote transpose of vector $(\alpha_{1r'},\cdots,\alpha_{nr'})$, where $\alpha_{ir'}$ is defined as in Table 5.2. The vector of technological coefficients for $u_{r'1}$ is:

$$\begin{bmatrix} h_{u_{r'1}}\\ Q\end{bmatrix}=\begin{bmatrix} -\alpha_{r'}\\ 0_{n\times 1}\\ 0_{n\times 1}\\ 0_{n_3\times 1}\\ 0_{n_3\times 1}\\ 0_{|R_1|\times 1}\\ 0_{|R_2|\times 1}\\ Q\end{bmatrix}$$

Assuming that $u_{r'1}$ is added to the basis, the matrix of technological coefficients for basic variables has the form:

$$A=\begin{bmatrix} B_{m\times m} & h_{u_{r'1}}\\ 0_{1\times m} & Q\end{bmatrix},$$

where the first n columns of A are associated with the variables in the optimal basis of the restricted master problem.

According to Proposition 3, the inverse of matrix A is calculated as:

$$A^{-1}=\begin{bmatrix} B^{-1} & -\frac{1}{Q}B^{-1}h_{u_{r'1}}\\ 0 & \frac{1}{Q}\end{bmatrix}$$

Then, the dual variables for all constraints, including the new constraint (5.23) to impose feasibility of route r', are determined by:

$$(\beta,\gamma,\delta,\rho,\eta,\theta,\pi,\theta_{r'}) = (c_B,\ c_{r'})\cdot\begin{bmatrix} B^{-1} & -\frac{1}{Q}B^{-1}h_{u_{r'1}} \\ 0 & \frac{1}{Q} \end{bmatrix}$$

As expected, the dual variables corresponding to the existing constraints do not change and are determined by:

$$(\beta,\gamma,\delta,\rho,\eta,\theta,\pi) = c_B B^{-1}.$$

The dual variable of the new constraint is determined as:

$$\theta_{r'} = \tfrac{1}{Q}(c_{r'} - c_B B^{-1} h_{u_{r'}}) = \tfrac{1}{Q}(c_{r'} + \textstyle\sum_{j\in r'}\beta_j).$$

$\square$

By extending the inverse matrix with column $u_{r'1}$, we can compute the reduced costs of variables $y_{ir'1}$ in column set $(r',1)$. Let $\varepsilon^i_{n\times 1}$ denote an $(n\times 1)$-unit vector with value 1 in position i. The vector of technological coefficients for variable $y_{ir'1}$ when $i\in r'\cap(V_{+0}\cup V_{0+})$ is:

$$\begin{bmatrix} h_{y_{ir'1}} \\ -1 \end{bmatrix} = \begin{bmatrix} 0_{n\times 1} \\ 0_{n\times 1} \\ \varepsilon^i_{n\times 1} \\ 0_{n_3\times 1} \\ 0_{n_3\times 1} \\ 0_{|R_1|\times 1} \\ 0_{|R_2|\times 1} \\ -1 \end{bmatrix}$$

So, the reduced cost of variable $y_{ir'1}$ for $i\in r'\cap V_{+0}$ is calculated as:

$$RC_{y_{ir'1}} = 0 - c_A A^{-1}\begin{bmatrix} h_{y_{ir'1}} \\ -1 \end{bmatrix} = -(\beta,\gamma,\delta,\rho,\eta,\theta,\pi,\theta_{r'})\cdot\begin{bmatrix} h_{y_{ir'1}} \\ -1 \end{bmatrix} = -\delta_i + \theta_{r'}$$
$$= -\delta_i + \frac{1}{Q}(\sum_{j\in r'}\beta_j + c_{r'})$$

And, the reduced cost of variable $y_{ir'1}$ for $i\in r'\cap V_{0+}$ is calculated as:

$$RC_{y_{ir'1}} = a_i - c_A A^{-1}\begin{bmatrix} h_{y_{ir'1}} \\ -1 \end{bmatrix} = a_i - (\beta,\gamma,\delta,\rho,\eta,\theta,\pi,\theta_{r'})\cdot\begin{bmatrix} h_{y_{ir'1}} \\ -1 \end{bmatrix}$$
$$= a_i - \delta_i + \theta_{r'} = a_i - \delta_i + \frac{1}{Q}(\sum_{j\in r'}\beta_j + c_{r'})$$

Similarly, the vector of technological coefficients for variable $y_{ir'1}$ when $i \in r' \cap V_{++}$ is:

$$\begin{bmatrix} h_{y_{ir'1}} \\ -1 \end{bmatrix} = \begin{bmatrix} 0_{n\times 1} \\ 0_{n\times 1} \\ \varepsilon^i_{n\times 1} \\ -\varepsilon^i_{n_3\times 1} \\ 0_{n_3\times 1} \\ 0_{|R_1|\times 1} \\ 0_{|R_2|\times 1} \\ -1 \end{bmatrix}$$

Hence, the reduced cost of variable $y_{ir'1}$ for $i \in r' \cap V_{++}$ is calculated as:

$$RC_{y_{ir'1}} = 0 - c_A A^{-1} \begin{bmatrix} h_{y_{ir'1}} \\ -1 \end{bmatrix} = -(\beta, \gamma, \delta, \rho, \eta, \theta, \pi, \theta_{r'}) \cdot \begin{bmatrix} h_{y_{ir'1}} \\ -1 \end{bmatrix}$$
$$= -\delta_i + \rho_i + \theta_{r'} = -\delta_i + \rho_i + \frac{1}{Q}(\sum_{j\in r'} \beta_j + c_{r'}).$$

Proposition 5. *Adding column set $(r', 1)$ to the master problem may improve the objective function if:*
$c_{r'} + \sum_{j\in r'} \beta_j - Q \max\{\max_{i\in(r'\cap V_{+0})}\{\delta_i\}, \max_{i\in(r'\cap V_{0+})}\{\delta_i - a_i\}, \max_{i\in(r'\cap V_{++})}\{\delta_i - \rho_i\}\} < 0$.

Similarly, adding column set $(r', 2)$ to the master problem may improve the objective function if:
$c_{r'} + \sum_{i\in r} \gamma_i - Q \max\{\max_{i\in(r\cap V_{+0})}\{\delta_i - p_i\}, \max_{i\in(r\cap V_{0+})}\{\delta_i\}, \max_{i\in(r\cap V_{++})}\{\delta_i - \eta_i\}\} < 0$.

Proof. We prove the proposition for column set $(r', 1)$. It is similar for column set $(r', 2)$. Assume that we add route r' to the master problem in iteration k of the Simplex algorithm. If in this iteration, $RC_{y_{ir'1}} < 0$ for at least some $i \in r'$, then having r' added to the master problem in iteration k may result in improving the objective value after $y_{ir'1}$ is added to the master problem. This implies that:

$\exists i \in (r' \cap V_{+0}) \mid c_{r'} + \sum_{j\in r'} \beta_j - Q(\delta_i) < 0$,
or $\exists i \in (r' \cap V_{0+}) \mid c_{r'} + \sum_{j\in r'} \beta_j - Q(\delta_i - a_i) < 0$,
or $\exists i \in (r' \cap V_{++}) \mid c_{r'} + \sum_{j\in r'} \beta_j - Q(\delta_i - \rho_i) < 0$.

The variable $y_{ir'1}$ most likely to have a negative reduced cost is the one with the largest value of δ_i (resp., $\delta_i - a_i$, $\delta_i - \rho_i$), for $i \in V_{+0}$ (resp., $i \in V_{0+}$, $i \in V_{++}$). Hence, the condition becomes:
$c_{r'} + \sum_{j\in r'} \beta_j - Q \max\{\max_{i\in(r'\cap V_{+0})}\{\delta_i\}, \max_{i\in(r'\cap V_{0+})}\{\delta_i - a_i\}, \max_{i\in(r'\cap V_{++})}\{\delta_i - \rho_i\}\} < 0$ □

5.3.3 Analysis of the dual master problem

We take another look at the previous reasoning, by analyzing the dual master problem. This approach is inspired from Le et al. [2013]. We know that if a solution is not optimal for the primal problem, then the associated dual solution is infeasible for the dual problem. More specifically, if there exists a variable in the primal minimization problem whose reduced cost is negative, then its associated constraint in the dual problem is violated. Route r in period t is associated with the column set (r,t), i.e., columns of technological coefficients of variables u_{rt} and y_{irt}, $\forall i \in r$. Therefore, adding column set (r,t) to the primal master problem may improve the objective function if any of the dual constraints associated with u_{rt} or y_{irt}'s are not respected. In order to see such dual constraints, hereunder we provide an explicit formulation of the dual of the master problem.

$$\max \quad -\sum_{i\in V}\beta_i - \sum_{i\in V}\gamma_i + \sum_{i\in V} d_i\delta_i - \sum_{i\in V_{++}} (d_{i1}\rho_i + d_{i2}\eta_i) \tag{5.25}$$

subject to

$$-\sum_{i\in r}\beta_i + Q\theta_r \le c_r;\ \forall r \in R_1,\ \text{(primal variables: } u_{r1}) \tag{5.26}$$

$$-\sum_{i\in r}\gamma_i + Q\pi_r \le c_r;\ \forall r \in R_2,\ \text{(primal variables: } u_{r2}) \tag{5.27}$$

$$\delta_i - \theta_r \le 0;\ \forall r \in R_1,\ i \in (r\cap V_{+0})\ \text{(primal variables: } y_{ir1}) \tag{5.28}$$

$$\delta_i - \theta_r \le a_i;\ \forall r \in R_1,\ i \in (r\cap V_{0+})\ \text{(primal variables: } y_{ir1}) \tag{5.29}$$

$$\delta_i - \rho_i - \theta_r \le 0;\ \forall r \in R_1,\ i \in (r\cap V_{++})\ \text{(primal variables: } y_{ir1}) \tag{5.30}$$

$$\delta_i - \pi_r \le p_i;\ \forall r \in R_2,\ i \in (r\cap V_{+0})\ \text{(primal variables: } y_{ir2}) \tag{5.31}$$

$$\delta_i - \pi_r \le 0;\ \forall r \in R_2,\ i \in (r\cap V_{0+})\ \text{(primal variables: } y_{ir2}) \tag{5.32}$$

$$\delta_i - \eta_i - \pi_r \le 0;\ \forall r \in R_2,\ i \in (r\cap V_{++})\ \text{(primal variables: } y_{ir2}) \tag{5.33}$$

$$\rho_i \le a_i;\ \forall i \in V_{++}\ \text{(primal variables: } w_{i1}) \tag{5.34}$$

$$\eta_i \le p_i;\ \forall i \in V_{++}\ \text{(primal variables: } w_{i2}) \tag{5.35}$$

$$\beta_i,\ \gamma_i,\ \delta_i,\ \rho_i,\ \eta_i,\ \theta_r,\ \pi_r \ge 0 \tag{5.36}$$

For any column set $(r,1)$, the corresponding constraints in the dual problem are:

$$\begin{cases} \theta_r \le \frac{c_r+\sum_{j\in r}\beta_j}{Q} & \\ \delta_i \le \theta_r & \forall i \in r\cap V_{+0} \\ \delta_i - a_i \le \theta_r & \forall i \in r\cap V_{0+} \\ \delta_i - \rho_i \le \theta_r & \forall i \in r\cap V_{++}. \end{cases} \tag{5.37}$$

Applying Fourier-Motzkin elimination on the set of Inequalities (5.37) results in:

$$\begin{cases} \delta_i \le \frac{c_r+\sum_{j\in r}\beta_j}{Q} & \forall i \in r\cap V_{+0} \\ \delta_i - a_i \le \frac{c_r+\sum_{j\in r}\beta_j}{Q} & \forall i \in r\cap V_{0+} \\ \delta_i - \rho_i \le \frac{c_r+\sum_{j\in r}\beta_j}{Q} & \forall i \in r\cap V_{++}. \end{cases} \tag{5.38}$$

We can say that at least one of the feasibility Inequalities (5.37) for the dual problem is violated if there exists an $i \in r$ such that:

$$\begin{cases} c_r + \sum_{j\in r} \beta_j - Q\delta_i < 0 & i \in r \cap V_{+0} \\ c_r + \sum_{j\in r} \beta_j - Q(\delta_i - a_i) < 0 & i \in r \cap V_{0+} \\ c_r + \sum_{j\in r} \beta_j - Q(\delta_i - \rho_i) < 0 & i \in r \cap V_{++}. \end{cases} \tag{5.39}$$

This is equivalent to saying that:

$$c_r + \sum_{j\in r} \beta_j - Q \max\{ \max_{i\in(r\cap V_{+0})} \{\delta_i\}, \max_{i\in(r\cap V_{0+})} \{\delta_i - a_i\}, \max_{i\in(r\cap V_{++})} \{\delta_i - \rho_i\}\} < 0 \tag{5.40}$$

By following the same calculations, one can say that column set $(r,2)$ may improve the objective function of the primal problem if:

$$c_r + \sum_{i\in r} \gamma_i - Q \max\{ \max_{i\in(r\cap V_{+0})} \{\delta_i - p_i\}, \max_{i\in(r\cap V_{0+})} \{\delta_i\}, \max_{i\in(r\cap V_{++})} \{\delta_i - \eta_i\}\} < 0 \tag{5.41}$$

We observe that Inequalities (5.40)–(5.41) are exactly the same criteria we obtained in Section 5.3.2 to identify the promising column sets to improve the objective function of the primal problem. The left hand side of Inequalities (5.40)–(5.41) can be regarded as two pricing problems to be minimized independently. If the optimal solution of both pricing problems result in nonnegative values for the objective functions (left hand sides), then the current solution of the master problem is optimal. Otherwise, we add to the master problem a set of column sets which result in a negative value for the objective function of the pricing problem. The next two sections illustrate how the pricing problems can be solved.

5.3.4 Pricing problems

We formulate a pricing problem for each period as an ESPPRC [Irnich and Desaulniers, 2005]. Each feasible solution of the ESPPRC is a route which starts and ends at the depot while including a subset of the vertices and respecting the side constraints related to elementarity and time windows. The settings are done in such a way that the cost of a route (solution) in the ESPPRC is equal to the objective function of the pricing problem, i.e., left hand side of Inequality (5.40) or (5.41). The objective function of the pricing problem in period 1 is written as:

$$\sum_{(i,j)\in A(r)} \bar{c}_{ij} - Q \max\{ \max_{i\in(r\cap V_{+0})} \{\delta_i\}, \max_{i\in(r\cap V_{0+})} \{\delta_i - a_i\}, \max_{i\in(r\cap V_{++})} \{\delta_i - \rho_i\}\}, \tag{5.42}$$

where the cost coefficients $\bar{c}_{ij}$'s are defined by Equation (5.43) and by convention $\beta_0 = 0$.

$$\bar{c}_{ij} = c_{ij} + \beta_j \tag{5.43}$$

Similarly, the objective function of the pricing problem in period 2 is written as:

$$\sum_{(i,j)\in A(r)} \bar{c}_{ij} - Q \max\{ \max_{i\in(r\cap V_{+0})} \{\delta_i - p_i\}, \max_{i\in(r\cap V_{0+})} \{\delta_i\}, \max_{i\in(r\cap V_{++})} \{\delta_i - \eta_i\}\}, \tag{5.44}$$

where the cost coefficients $\bar{c}_{ij}$'s are defined by Equation (5.45) and by convention $p_0 = 0$ and $\beta_0 = 0$.

$$\bar{c}_{ij} = c_{ij} + \gamma_j \tag{5.45}$$

For each period 1 and 2, we set up a distinct network on the vertex set V^0, where the cost $\bar{c}_{ij}$ of each arc (i,j) is given either by (5.43) or by (5.45), depending on the period. Other parameters of the network are listed in Table 5.2.

5.3.5 The label-setting algorithm

A label-setting algorithm is used to solve the pricing problems (ESPPRC). In this algorithm, a multi-dimensional label L_i is associated with each path from the depot to an end vertex i. In expanded form, the components of label L_i are $(L_i^{cost}, L_i^{dual}, L_i^{time}, (L_i^k)_{k\in V})$, where each component indicates the consumption of a limited resource along the path with which L_i is associated. The path under consideration is feasible if all components of L_i respect the limits on available resources.

The first component L_i^{cost} denotes the sum of $\bar{c}_{ij}$'s over the arcs (i,j) covered by the path, where $\bar{c}_{ij}$ is calculated by Equations (5.43) and (5.45) for periods 1 and 2, respectively, and there is no resource constraint on it. The second component L_i^{dual} is a term we need to keep track of the coefficient of Q in Equations (5.42) and (5.44), and there is no resource constraint on it. The third component L_i^{time} shows the time when store i is visited and the service starts at this store. It must respect the time window for store i, that is,

$$e_i \leq L_i^{time} \leq l_i. \tag{5.46}$$

Finally, L_i^k indicates the number of times store k is visited by path L_i. Each path must be elementary, meaning that in each period a path cannot visit any store k more than once:

$$0 \leq L_i^k \leq 1 \quad \text{for all } k \in V. \tag{5.47}$$

The label-setting algorithm starts from the initial label $L_0 = (0, 0, e_0, (0)_{k\in V})$, associated with the depot, and generates new labels using extension functions. A label L_i is extended along all arcs $(i,j) \in A$ and new labels $L_j = (L_j^{cost}, L_j^{dual}, L_j^{time}, (L_j^k)_{k\in V})$ are created. Define $\bar{\delta}_i$ for periods 1 and 2 by Equations (5.48) and (5.49), respectively.

$$\bar{\delta}_i = \begin{cases} \delta_j & j \in V_{+0} \\ \delta_j - a_j & j \in V_{0+} \\ \delta_j - \rho_j & j \in V_{++} \end{cases} \tag{5.48}$$

$$\bar{\delta}_i = \begin{cases} \delta_j - p_i & j \in V_{+0} \\ \delta_j & j \in V_{0+} \\ \delta_j - \eta_j & j \in V_{++} \end{cases} \tag{5.49}$$

Then, the functions extending label L_i to label L_j along arc (i,j) are as follows:

$$L_j^{cost} = L_i^{cost} + \bar{c}_{ij},$$
$$L_j^{dual} = \max\{L_i^{dual}, \bar{\delta}_j\},$$
$$L_j^{time} = \max\{e_j, L_i^{time} + s_i + t_{ij}\},$$
$$L_j^k = \begin{cases} L_j^k + 1 & \text{if } k = j \\ L_j^k & \text{otherwise.} \end{cases}$$

A label L_j is discarded if at least one of its resource components exceeds the corresponding limits in inequalities (5.46)-(5.47). A feasible route is constructed by extending a feasible path to the depot, provided that the extended label to the depot remains feasible.

To avoid enumerating all feasible paths, a dominance rule is applied to eliminate labels that are not Pareto optimal and, therefore, cannot yield an optimal path [Gutierrez-Jarpa et al., 2010]. Given two labels $(L_j)_1 = (L_j^{cost}, L_j^{dual}, L_j^{time}, (L_j^k)_{k\in V})_1$ and $(L_j)_2 = (L_j^{cost}, L_j^{dual}, L_j^{time}, (L_j^k)_{k\in V})_2$ ending at the same vertex j, this rule stipulates that $(L_j)_1$ dominates $(L_j)_2$ if we have: $(L_j^{time})_1 \le (L_j^{time})_2$, $(L_j^k)_1 \le (L_j^k)_2 \ \ \forall k \in V$, $(L_j^{cost} - QL_j^{dual})_1 \le (L_j^{cost} - QL_j^{dual})_2$, and the inequality is strict for at least one component.

Proposition 6. *If $RC_{y_{irt}} > 0$ in path r in period t, then $RC_{y_{ir'1}} > 0$ in all extended paths r' from r in period t.*

Proof. We have $RC_{y_{irt}} = -\bar{\delta}_i + \frac{1}{Q}\sum_{(k,j)\in A(r)} \bar{c}_{kj}$, where $\bar{c_{kj}}$ and $\bar{\delta}_i$ are calculated by Equations (5.43) and (5.48) when $t = 1$ and by Equations (5.45) and (5.49) when $t = 2$. Assume that path r ending at store j is extended to another path r' including a new store j'. We obtain:

$$RC_{y_{ir't}} = -\bar{\delta}_i + \tfrac{1}{Q}\textstyle\sum_{(k,j)\in r} \bar{c}_{kj} + \tfrac{1}{Q}\bar{c}_{jj'} > -\bar{\delta}_i + \tfrac{1}{Q}\sum_{(k,j)\in r} \bar{c}_{kj} = RC_{y_{irt}} > 0 \qquad \square$$

Proposition 7. *The current solution of the master problem is optimal if the following two conditions hold simultaneously:*
1) $c_{0i} + c_{i0} + \beta_i - Q\bar{\delta}_i \ge 0$, $\forall i \in V$, where $\bar{\delta}_i$ is determined by Equation (5.48),
2) $c_{0i} + c_{i0} + \gamma_i - Q\bar{\delta}_i \ge 0$, $\forall i \in V$, where $\bar{\delta}_i$ is determined by Equation (5.49).

Proof. Consider an arbitrary route r in period 1 which includes store i along with a set of other stores. We have $RC_{y_{ir1}} = c_r + \beta_i - Q\bar{\delta}_i \ge c_{0i} + c_{i0} + \beta_i - Q\bar{\delta}_i \ge 0$. Similarly, for any arbitrary route in period 2 which includes store i we have $RC_{y_{ir2}} = c_r + \gamma_i - Q\bar{\delta}_i \ge c_{0i} + c_{i0} + \gamma_i - Q\bar{\delta}_i \ge 0$.

This implies that if $c_{0i} + c_{i0} + \beta_i - Q\bar{\delta}_i \ge 0$, then $RC_{y_{ir1}} \ge 0$ in any arbitrary route in period 1. Similarly, if $c_{0i} + c_{i0} + \gamma_i - Q\bar{\delta}_i \ge 0$, then $RC_{y_{ir2}} \ge 0$ in any arbitrary route in period 2. When these two conditions hold for all stores, it implies that for any arbitrary route in period 1 or 2, $RC_{y_{irt}} \ge 0$ for every $i \in V$. So, the current optimal solution to the restricted master problem is optimal to the master problem. $\square$

The same acceleration techniques discussed in Section 4.3.5 are applicable here, i.e., bidirectional search, inaccessible vertices, and relaxed dominance rule. The last two techniques are exactly the same as we described in Section 4.3.5. However, applying the bidirectional search needs a bit more attention.

If L_i is a forward label associated with a path $(0,\ldots,i)$ ending at vertex i, and Γ_j is a backward label associated with a path $(j,\ldots,0)$ starting from vertex j, then concatenating the two paths yields a complete route $(0,\ldots,0)$ with cost $L_i^{cost} + \Gamma_j^{cost} + \bar{c}_{ij} - Q \cdot \max\{L_i^{dual}, \Gamma_j^{dual}\}$. The route is feasible if the following conditions hold:
$L_i^{time} + s_i + t_{ij} \le \Gamma_j^{time}$,
$L_i^k + \Gamma_j^k \le 1$ for all $k \in V$.

5.4 Column generation

In this section, a different formulation is proposed for the two-period VRP with partial delivery shifts. This formulation is inspired by the work of Desaulniers [2010] on VRP with split deliveries. As opposed to model (5.1)-(5.11), the variables in this model simultaneously represent decisions about the choice of routes and about quantities to be delivered along these routes. We will also sketch a column generation algorithm which can be used to solve its LP-relaxation. The pricing problem is an ESPPRC combined with the linear relaxation of a bounded knapsack problem.

5.4.1 A mixed integer linear programming formulation

Let us consider an arbitrary elementary route r that respects time windows at all stores and at the depot. We define the set of *feasible delivery pattern* for route r as the bounded polyhedron

$$\Omega_r = \{y = (y_i)_{i\in r} | \sum_{i\in r} y_i \leq Q,\ 0 \leq y_i \leq d_i \text{ for all } i \in r\}.$$

Since Ω_r is a bounded knapsack polyhedron, its extreme points are exactly the vectors $y \in \Omega_r$ such that $y_i \in \{0, d_i\}$ for all stores $i \in r$, except possibly one store j for which $0 < y_j < d_j$. We denote these extreme delivery patterns as y_r^k, $k \in K_r$. All feasible delivery patterns in Ω_r are convex combinations of the extreme ones, that is, a vector y is in Ω_r if and only if there exist nonnegative multipliers u^k, $k \in K_r$, such that $y = \sum_{k\in K_r} y_r^k u^k$ and $\sum_{k\in K_r} u^k = 1$.

The new formulation accordingly rests on decision variables u_{rt}^k, $k \in K_r$, which can be intuitively interpreted as the fraction of route $r \in R_t$ used in period t with the extreme delivery pattern k (see Table 5.4).

Table 5.4: Sets, parameters, and decision variables

K_r	set of feasible extreme delivery patterns of route r
y_{ir}^k	delivery quantity to store i in period t based on extreme delivery pattern k
u_{rt}^k	fraction of route $r \in R_t$ used in period t with extreme delivery pattern $k \in K_r$

Consider now the integer linear programming problem (5.50)-(5.56) hereunder:

$$\begin{aligned} \min & \sum_{r\in R_1}\sum_{k\in K_r} c_r u_{r1}^k + \sum_{r\in R_2}\sum_{k\in K_r} c_r u_{r2}^k \\ & + \sum_{r\in R_2}\sum_{i\in(r\cap V_{+0})} p_i \sum_{k\in K_r} y_{ir}^k u_{r2}^k + \sum_{r\in R_1}\sum_{i\in(r\cap V_{0+})} a_i \sum_{k\in K_r} y_{ir}^k u_{r1}^k \\ & + \sum_{i\in V_{++}} (a_i w_{i1} + p_i w_{i2}) \end{aligned} \tag{5.50}$$

subject to

$$\sum_{r\in R_1}\sum_{k\in K_r} \alpha_{ir} u_{r1}^k \leq 1 \quad \forall i \in V \tag{5.51}$$

$$\sum_{r\in R_2}\sum_{k\in K_r}\alpha_{ir}u^k_{r2}\le 1 \quad \forall i\in V \tag{5.52}$$

$$\sum_{r\in R_1}\sum_{k\in K_r}y^k_{ir}u^k_{r1}+\sum_{r\in R_2}\sum_{k\in K_r}y^k_{ir}u^k_{r2}=d_i \quad \forall i\in V \tag{5.53}$$

$$\sum_{r\in R_1}\sum_{k\in K_r}y^k_{ir}u^k_{r1}-w_{i1}\le d_{i1} \quad \forall i\in V_{++} \tag{5.54}$$

$$\sum_{r\in R_2}\sum_{k\in K_r}y^k_{ir}u^k_{r2}-w_{i2}\le d_{i2} \quad \forall i\in V_{++} \tag{5.55}$$

$$u^k_{rt},\ w_{it}\ge 0 \tag{5.56}$$

$$\sum_{k\in K_r}u^k_{rt}\in\{0,1\} \quad \forall r\in R_t \text{ and for } t=1,2 \tag{5.57}$$

In order to argue that (5.50)-(5.57) provides a correct formulation of the two-period VRP with partial delivery shifts, let us consider an arbitrary feasible solution of this model, say, (u^k_{rt},w_{it}) ($t=1,2$; $r\in R_t$; $k\in K_r$; $i\in r$), and let us introduce the following auxiliary quantities which can be compared with those defined in Table 5.3. First, for all $r\in R_t$ and for $t=1,2$, let

$$u_{rt}=\sum_{k\in K_r}u^k_{rt}. \tag{5.58}$$

Note that this is a binary variable by virtue of constraint (5.57). As in Table 5.3, our interpretation is that $u_{rt}=1$ if route r is used in period t. Next, for $i\in r$, let us interpret

$$y_{irt}=\sum_{k\in K_r}y^k_{ir}u^k_{rt} \tag{5.59}$$

as the quantity delivered to store i along route r in period t.

We claim that (u_{rt},y_{irt},w_{it}) defines a feasible solution of model (5.1)-(5.11) with the same cost as (u^k_{rt},w_{it}) for model (5.50)-(5.57). Indeed, substituting (5.58) and (5.59) in the objective function (5.50) and in the constraints (5.51)-(5.57), it is obvious that the solution (u_{rt},y_{irt},w_{it}) yields the same value of (5.1) as (5.50), and that it satisfies (5.2)-(5.6), (5.9)-(5.11). Moreover, constraints (5.7)-(5.8) are also satisfied since, for every route r and for $t=1,2$,

$$\sum_{i\in r}y_{irt}=\sum_{i\in r}\sum_{k\in K_r}y^k_{ir}u^k_{rt}\le Q\sum_{k\in K_r}u^k_{rt}=Qu_{rt},$$

where the inequality holds because $y^k_r\in\Omega_r$.

So, we have shown that, for every feasible solution (u^k_{rt},w_{it}) of model (5.51)-(5.57), there exists a feasible solution (u_{rt},y_{irt},w_{it}) of (5.2)-(5.11) with the same cost.

Conversely, consider now an arbitrary feasible solution of (5.2)-(5.11), say (u_{rt},y_{irt},w_{it}). For each t and $r\in R_t$, the vector $y_{rt}=(y_{irt})_{i\in r}$ is in Ω_r. If $u_{rt}=0$ then we let $u^k_{rt}=0$ for all $k\in K_r$. Otherwise, $u_{rt}=1$ and we write y_{rt} as a convex combination of extreme patterns, say $y_{rt}=\sum_{k\in K_r}y^k_r u^k_{rt}$ with $\sum_{k\in K_r}u^k_{rt}=1$.

With these definitions, equations (5.58) and (5.59) hold, and it is easy to check that (u^k_{rt},w_{it}) is a feasible solution of model (5.51)-(5.56), with the same cost as (u_{rt},y_{irt},w_{it}).

So, we conclude that the solutions of (5.2)-(5.11) and of (5.51)-(5.57) are in one-to-one relation through a cost-preserving mapping. This implies, in particular, that (5.50)-(5.57) is a valid formulation of the two-period VRP with partial delivery shifts. Note that constraints (5.53) can be replaced by constraints (5.60) without impacting the optimal solution.

$$\sum_{r\in R_1}\sum_{k\in K_r} y_{ir}^k u_{r1}^k + \sum_{r\in R_2}\sum_{k\in K_r} y_{ir}^k u_{r2}^k \geq d_i;\ \forall i \in V \tag{5.60}$$

5.4.2 Master problem

Consider the LP-relaxation of Problem (5.50)-(5.57) where Constraints (5.53) are replaced by Constraints (5.60), Constraints (5.57) are eliminated, as they are satisfied by Constraints (5.51)-(5.52), and other constraints are transformed into the form of $\geq$ so that all the corresponding dual variables are non-negative. Problem (5.61)-(5.67) is the resultant master problem.

$$\begin{aligned}
&\min \sum_{r\in R_1}\sum_{k\in K_r} c_r u_{r1}^k + \sum_{r\in R_2}\sum_{k\in K_r} c_r u_{r2}^k \\
&+ \sum_{r\in R_2}\sum_{i\in(r\cap V_{+0})} p_i \sum_{k\in K_r} y_{ir}^k u_{r2}^k + \sum_{r\in R_1}\sum_{i\in(r\cap V_{0+})} a_i \sum_{k\in K_r} y_{ir}^k u_{r1}^k \\
&+ \sum_{i\in V_{++}} (a_i w_{i1} + p_i w_{i2})
\end{aligned} \tag{5.61}$$

subject to

$$-\sum_{r\in R_1}\sum_{k\in K_r} \alpha_{ir} u_{r1}^k \geq -1 \ \ \forall i\in V \ \text{(dual variables: } \beta_i) \tag{5.62}$$

$$-\sum_{r\in R_2}\sum_{k\in K_r} \alpha_{ir} u_{r2}^k \geq -1 \ \ \forall i\in V \ \text{(dual variables: } \gamma_i) \tag{5.63}$$

$$\sum_{r\in R_1}\sum_{k\in K_r} y_{ir}^k u_{r1}^k + \sum_{r\in R_2}\sum_{k\in K_r} y_{ir}^k u_{r2}^k \geq d_i \ \ \forall i\in V \ \text{(dual variables: } \delta_i) \tag{5.64}$$

$$-\sum_{r\in R_1}\sum_{k\in K_r} y_{ir}^k u_{r1}^k + w_{i1} \geq -d_{i1} \ \ \forall i\in V_{++} \ \text{(dual variables: } \rho_i) \tag{5.65}$$

$$-\sum_{r\in R_2}\sum_{k\in K_r} y_{ir}^k u_{r2}^k + w_{i2} \geq -d_{i2} \ \ \forall i\in V_{++} \ \text{(dual variables: } \eta_i) \tag{5.66}$$

$$u_{rt}^k,\ w_{it} \geq 0 \tag{5.67}$$

Note that in the LP-relaxation of model (5.50)-(5.57) there is no need to keep constraints (5.57) anymore. Indeed, constraints (5.51)-(5.52) imply that $\sum_{k\in K_r} u_{rt}^k \leq 1$ holds for all $r \in R_t$ and for $t = 1,2$.

Although ILP models (5.1)-(5.11) and (5.50)-(5.57) are equivalent, their linear relaxations (5.15)-(5.24) and (5.61)-(5.67) *are not*. Actually, consider first a feasible solution (u_{rt}^{k}, w_{it}) of the linear relaxation (5.62)-(5.67). The same reasoning as in Section 5.4.1 shows that the solution $(u_{rt}, y_{irt}, w_{it})$ defined by (5.58)-(5.59) is feasible for (5.16)-(5.24), with the same cost. This shows that the lower bound obtained by solving the linear relaxation of (5.1)-(5.11) is at least as large as the lower bound obtained from the linear relaxation of (5.50)-(5.57). But the converse relation does not hold anymore, as evidenced by the following example.

Example. Assume there is a single store 1 (hence, a single route r) with demand $d_{11} = 3$ in period 1 and no demand in period 2. The cost of visiting the store is $c_r = 7$. The vehicle capacity is $Q = 10$. Assume also that the postponement penalties are large, so that the store is necessarily visited in period 1 in any optimal solution. The linear relaxation of the first model has the optimal solution $y_1 = 3$, and $u_r = 0.3$ with cost 2.1. For the second model, Ω_r is the interval [0,3] with extreme points $y_r^1 = 0$ and $y_r^2 = 3$. The linear relaxation of this model has the optimal solution $(u_r^1, u_r^2) = (0, 1)$, with cost 7.

5.4.3 Pricing problems

Consider a new extreme delivery pattern k of (new) route r in period 1. The reduced cost of variable u_{r1}^{k} is determined as:

$$c_r + \sum_{i\in(r\cap V_{0+})} a_i y_{ir}^k - \left(-\sum_{i\in r}\beta_i + \sum_{i\in r} y_{ir}^k \delta_i - \sum_{i\in(r\cap V_{++})} y_{ir}^k \rho_i \right) = \sum_{(i,j)\in r} c_{ij}$$
$$+ \sum_{i\in(r\cap V_{+0})} (\beta_i - \delta_i y_{ir}^k) + \sum_{i\in(r\cap V_{0+})} (\beta_i + a_i y_{ir}^k - \delta_i y_{ir}^k) + \sum_{i\in(r\cap V_{++})} (\beta_i + \rho_i y_{ir}^k - \delta_i y_{ir}^k) \tag{5.68}$$

Let us define $\bar{c}_{ij}$ and $\bar{\delta}_j$ as follows:

$$\bar{c}_{ij} = c_{ij} + \beta_j \tag{5.69}$$

$$\bar{\delta}_j = \begin{cases} \delta_j & j \in V_{+0} \\ \delta_j - a_j & j \in V_{0+} \\ \delta_j - \rho_j & j \in V_{++} \end{cases} \tag{5.70}$$

The reduced cost of variable u_{r1}^{k} in Equation (5.68) can be rewritten as:

$$\sum_{(i,j)\in r} \bar{c}_{ij} - \sum_{i\in r} y_{ir}^k \bar{\delta}_i \tag{5.71}$$

Following the same calculations for a new pattern of a (new) route in period 2 leads to the same formula as Equation (5.71), where $\bar{\delta}_j$ is determined by:

$$\bar{\delta}_j = \begin{cases} \delta_j - p_i & j \in V_{+0} \\ \delta_j & j \in V_{0+} \\ \delta_j - \eta_j & j \in V_{++} \end{cases} \tag{5.72}$$

In order to find a new extreme delivery pattern k for a (new) route with the lowest reduced cost in period t, we need to solve the following problem:

$$\min \sum_{(i,j)\in r} \bar{c}_{ij} - \sum_{i\in r} y^k_{ir}\bar{\delta}_i \tag{5.73}$$

subject to

$$\sum_{i\in r} y^k_{ir} \le Q \tag{5.74}$$

$$0 \le y^k_{ir} \le d_i; \quad \forall i \in r \tag{5.75}$$

$$r: \text{ is elementary and respects time windows} \tag{5.76}$$

5.4.4 The label-setting algorithm

As in Desaulniers [2010], a label-setting algorithm can be used to solve problem (5.73)-(5.76). For each path, we define a label L_i whose components in the expanded form are $(L_i^{cost}, L_i^{time}, L_i^{load}, (L_i^k)_{k\in V}, L_i^{partial}, L_i^{max}, L_i^{dual})$, where each component is defined as follows:

- L_i^{cost} : The value of the objective function (5.73), where the store with a partial delivery is neglected in the second term, i.e., $L_i^{cost} = \sum_{(i,j)\in r} \bar{c}_{ij} - \sum_{i\in r|y^k_{ir}\in\{0,d_i\}} y^k_{ir}\bar{\delta}_i$. There is no resource constraint on the cost component.
- L_i^{time} : The time when store i is visited and the service starts at this store. It must respect the time window for store i, i.e., $e_i \le L_i^{time} \le l_i$.
- L_i^{load} : The load on the vehicle, where the partial delivery is neglected, i.e., $L_i^{load} = \sum_{i\in r|y^k_{ir}\in\{0,d_i\}} y^k_{ir}$.
- L_i^k : The number of times store k is visited by the path corresponding to label L_i. For an elementary path, each store can be visited at most once, i.e., $0 \le L_i^k \le 1$ for all $k \in V$.
- $L_i^{partial}$: The number of partial deliveries in the path. $L_i^{partial} \in \{0,1\}$ is the resource constraint on it.
- L_i^{max} : It is equal to zero if $L_i^{partial} = 0$. Otherwise, it shows the maximum quantity that can be delivered to the store with a partial delivery in the path. There is no resource constraint on it.
- L_i^{dual} : It is equal to zero if $L_i^{partial} = 0$. Otherwise, it gives the modified unit dual price of the store with a partial delivery, i.e., $\bar{\delta}_{i'}$ where i' is the index of the store with a partial delivery. There is no resource constraint on it.

The reduced cost of a path with label L_i ending at store i is a linear function of the total quantity, say q, delivered to all stores visited by the path (including the store with a partial delivery), and can be written as: $Z(q) = L_i^{cost} - (q - L_i^{load})L_i^{dual}$, where $q \in [L_i^{load}, L_i^{load} + L_i^{max}]$. In this sense, when $L_i^{dual} \ge 0$, the best value for a partial delivery resulting in the lowest reduced cost is L_i^{max} which leads to $Z(L_i^{load} + L_i^{max}) = L_i^{cost} - L_i^{max}L_i^{dual}$. When $L_i^{dual} < 0$, the best value for the partial delivery is zero which leads to the lowest reduced cost as $Z(L_i^{load}+0) = L_i^{cost}$. However, when store i' is considered to be delivered partially in a path, we only let delivering the maximum possible quantity to i' because a zero delivery to it is performed in other labels.

Given a feasible label L_i ending at store i associated with an extreme delivery pattern, the label can be extended to store j up to three times along arc (i,j); one for a zero delivery to store j, one for a partial delivery $d_{jt}^k < d_j$, and one for a full delivery d_j. These three cases are discussed hereunder.

Case 1 (Zero delivery): If $L_i^{time} + s_i + t_{ij} \leq l_j$ and $L_i^j = 0$, then a zero delivery can occur at j to create a new label L_{jZ}. Components of the extended label are calculated as follows:

$$
\begin{aligned}
&L_{jZ}^{cost} = L_i^{cost} + \bar{c}_{ij},\\
&L_{jZ}^{load} = L_i^{load},\\
&L_{jZ}^{time} = \max\{L_i^{time} + s_i + t_{ij}, e_j\},\\
&L_{jZ}^{k} = \begin{cases} L_i^k + 1 & \text{if } k = j \\ L_i^k & \text{otherwise,} \end{cases}\\
&L_{jZ}^{partial} = L_i^{partial},\\
&L_{jZ}^{max} = L_i^{max},\\
&L_{jZ}^{dual} = L_i^{dual}.
\end{aligned}
$$

Case 2 (Partial delivery): If $L_i^{time} + s_i + t_{ij} \leq l_j$, $L_i^j = 0$, and $L_i^{partial} = 0$, then a partial delivery can occur at j to create a new label L_{jP}. In the extended label, components are calculated as follows:

$$
\begin{aligned}
&L_{jP}^{cost} = L_i^{cost} + \bar{c}_{ij},\\
&L_{jP}^{load} = L_i^{load},\\
&L_{jP}^{time} = \max\{L_i^{time} + s_i + t_{ij}, e_j\},\\
&L_{jP}^{k} = \begin{cases} L_i^k + 1 & \text{if } k = j \\ L_i^k & \text{otherwise,} \end{cases}\\
&L_{jP}^{partial} = L_i^{partial} + 1,\\
&L_{jP}^{max} = \min\{d_j, Q - L_i^{load}\},\\
&L_{jP}^{dual} = \bar{\bar{\delta}}_j.
\end{aligned}
$$

Case 3 (Full delivery): If $L_i^{time} + s_i + t_{ij} \leq l_j$, $L_i^j = 0$, and $L_i^{load} + d_j \leq Q$, then a full delivery can occur at j to create a new label L_{jF}. In the extended label, components are computed as follows:

$$
\begin{aligned}
&L_{jF}^{cost} = L_i^{cost} + \bar{c}_{ij} - d_j\bar{\bar{\delta}}_j,\\
&L_{jF}^{load} = L_i^{load} + d_j,\\
&L_{jF}^{time} = \max\{L_i^{time} + s_i + t_{ij}, e_j\},\\
&L_{jF}^{k} = \begin{cases} L_i^k + 1 & \text{if } k = j \\ L_i^k & \text{otherwise,} \end{cases}\\
&L_{jF}^{partial} = L_i^{partial},\\
&L_{jF}^{max} = \min\{L_i^{max}, Q - L_i^{load} - d_j\},\\
&L_{jF}^{dual} = L_i^{dual}.
\end{aligned}
$$

A domination rule is applied based on the following proposition.

Proposition 8. *Given two labels $(L_j)_1$ and $(L_j)_2$ ending at the same vertex j, $(L_j)_1$ dominates $(L_j)_2$ if all the following criteria hold, where $(x)^+ = \max\{0, x\}$.*

1. $L_{j1}^{load} \leq L_{j2}^{load}$,
2. $L_{j1}^{time} \leq L_{j2}^{time}$,
3. $L_{j1}^{k} \leq L_{j2}^{k} \;\; \forall k \in V$,
4. $L_{j1}^{partial} \leq L_{j2}^{partial}$,
5. $L_{j1}^{cost} - L_{j1}^{max} \cdot (L_{j1}^{dual})^+ \leq L_{j2}^{cost} - L_{j2}^{max} \cdot (L_{j2}^{dual})^+$,
6. $L_{j1}^{cost} - (L_{j2}^{load} - L_{j1}^{load}) \cdot (L_{j1}^{dual})^+ \leq L_{j2}^{cost}$,
7. $L_{j1}^{cost} - (L_{j2}^{load} - L_{j1}^{load} + L_{j2}^{max}) \cdot (L_{j1}^{dual})^+ \leq L_{j2}^{cost} - L_{j2}^{max} \cdot (L_{j2}^{dual})^+$.

Proof. Given two labels $(L_j)_1$ and $(L_j)_2$ ending at the same vertex j, $(L_j)_1$ dominates $(L_j)_2$ if two conditions are held: (1) all feasible extensions of $(L_j)_2$ are also feasible for $(L_j)_1$ and (2) the reduced cost of every feasible path extended from $(L_j)_2$ to an arbitrary vertex is greater than or equal to that of a feasible path extended from $(L_j)_1$ to the same vertex.

The first condition translates into criteria 1-4. In order to analyze the second condition, assume that labels L_{j1} and L_{j2} ending at the same vertex j are given and criteria 1-4 are satisfied. Assume that we extend both labels to a new vertex k. Based on the definition of an extreme delivery pattern, the delivery quantity to store k can be zero, partial (given that $L_{j1}^{partial} = L_{j2}^{partial} = 0$), or full. $(L_j)_1$ dominates $(L_j)_2$ if for any type of delivery to the arbitrary store k the cost of a new label extending from $(L_j)_1$ to k is less than or equal to the cost of a new label extending from $(L_j)_2$ to k given that the delivery quantity to k is zero, partial, or full for both the extended labels. Here, we analyze these three cases.

Proof of Case 1 (zero delivery to k): The best cost of the extended labels from $(L_j)_1$ and $(L_j)_2$ to k are compared based on Inequality (5.77), where $(L_k)_1$ and $(L_k)_2$ are the new extended labels.

$$\begin{aligned} &\min_{q \in [L_{k1}^{load}, L_{k1}^{load} + L_{k1}^{max}]} L_{k1}^{cost} - (q - L_{k1}^{load}) \cdot L_{k1}^{dual} \leq \\ &\min_{q \in [L_{k2}^{load}, L_{k2}^{load} + L_{k2}^{max}]} L_{k2}^{cost} - (q - L_{k2}^{load}) \cdot L_{k2}^{dual} \end{aligned} \tag{5.77}$$

Since the delivery quantity is zero, we can plug in the values of $(L_k)_1$ and $(L_k)_2$ based on Case 1 of the label-setting algorithm, and we obtain Inequality (5.78).

$$\begin{aligned} &\min_{q \in [L_{j1}^{load}, L_{j1}^{load} + L_{j1}^{max}]} L_{j1}^{cost} + \bar{c}_{jk} - (q - L_{j1}^{load}) \cdot L_{j1}^{dual} \leq \\ &\min_{q \in [L_{j2}^{load}, L_{j2}^{load} + L_{j2}^{max}]} L_{j2}^{cost} + \bar{c}_{jk} - (q - L_{j2}^{load}) \cdot L_{j2}^{dual} \end{aligned} \tag{5.78}$$

The minimization problems on two sides of Inequality (5.78) can conveniently be solved. Plugging in the best value of q, we obtain the following condition which is exactly criterion 5 in Proposition 8.

$$L_{j1}^{cost} - L_{j1}^{max} \cdot (L_{j1}^{dual})^{+} \le L_{j2}^{cost} - L_{j2}^{max} \cdot (L_{j2}^{dual})^{+} \tag{5.79}$$

Proof of Case 2 (partial delivery to k): When store k receives a partial delivery, the elements of labels $(L_k)_1$ and $(L_k)_2$ can be replaced in Inequality (5.77) by the values defined in Case 2 of the label-setting algorithm. This results in the following inequality.

$$\begin{aligned} &\min_{q \in [L_{j1}^{load}, L_{j1}^{load} + \min\{d_k, Q - L_{j1}^{load}\}]} L_{j1}^{cost} + \bar{c}_{jk} - (q - L_{j1}^{load}) \cdot \bar{\bar{\delta}}_k \le \\ &\min_{q \in [L_{j2}^{load}, L_{j2}^{load} + \min\{d_k, Q - L_{j2}^{load}\}]} L_{j2}^{cost} + \bar{c}_{jk} - (q - L_{j2}^{load}) \cdot \bar{\bar{\delta}}_k \end{aligned} \tag{5.80}$$

The optimal values of the minimization problems in Inequality (5.80) are easily determined and the inequality to be established is simplified as follows:

$$L_{j1}^{cost} - \min\{d_k, Q - L_{j1}^{load}\} \cdot (\bar{\bar{\delta}}_k)^{+} \le L_{j2}^{cost} - \min\{d_k, Q - L_{j2}^{load}\} \cdot (\bar{\bar{\delta}}_k)^{+} \tag{5.81}$$

Now, from $L_{j1}^{load} \le L_{j2}^{load}$ (criterion 1), it trivially follows that

$$\min\{d_k, Q - L_{j1}^{load}\} \ge \min\{d_k, Q - L_{j2}^{load}\}. \tag{5.82}$$

Moreover, note that extensions with partial deliveries to k are possible only if $L_{j1}^{partial} = L_{j2}^{partial} = 0$, which implies $L_{j1}^{dual} = L_{j2}^{dual} = 0$. Then, criterion 5 implies that $L_{j1}^{cost} \le L_{j2}^{cost}$. From this inequality and from (5.82), we get (5.81), as required.

Proof of Case 3 (full delivery to k): When store k receives a full delivery, the elements of labels $(L_k)_1$ and $(L_k)_2$ based on Case 3 of the label-setting algorithm should be plugged in Inequality (5.77). This results in the following inequality:

$$\begin{aligned} &\min_{q \in [L_{j1}^{load} + d_k, L_{j1}^{load} + d_k + \min\{L_{j1}^{max}, Q - L_{j1}^{load} - d_k\}]} L_{j1}^{cost} + \bar{c}_{jk} - d_k \cdot \bar{\bar{\delta}}_k - (q - L_{j1}^{load} - d_k) \cdot L_{j1}^{dual} \le \\ &\min_{q \in [L_{j2}^{load} + d_k, L_{j2}^{load} + d_k + \min\{L_{j2}^{max}, Q - L_{j2}^{load} - d_k\}]} L_{j2}^{cost} + \bar{c}_{jk} - d_k \cdot \bar{\bar{\delta}}_k - (q - L_{j2}^{load} - d_k) \cdot L_{j2}^{dual} \end{aligned} \tag{5.83}$$

The optimal values of the minimization problems in Inequality (5.83) are conveniently calculated and the inequality is simplified as follows:

$$\begin{aligned} &L_{j1}^{cost} - \min\{L_{j1}^{max}, Q - L_{j1}^{load} - d_k\} \cdot (L_{j1}^{dual})^{+} \le \\ &L_{j2}^{cost} - \min\{L_{j2}^{max}, Q - L_{j2}^{load} - d_k\} \cdot (L_{j2}^{dual})^{+} \end{aligned} \tag{5.84}$$

Let us now consider three distinct subcases.

Subcase 3-1: $L_{j1}^{max} \le Q - L_{j1}^{load} - d_k$. In this case, Inequality (5.84) simplifies to

$$L_{j1}^{cost} - L_{j1}^{max} \cdot (L_{j1}^{dual})^{+} \le L_{j2}^{cost} - \min\{L_{j2}^{max}, Q - L_{j2}^{load} - d_k\} \cdot (L_{j2}^{dual})^{+}. \tag{5.85}$$

By criterion 5 of Proposition 8, we straightforwardly draw Inequality (5.85) as follows:

$$\begin{aligned} &L_{j1}^{cost} - L_{j1}^{max} \cdot (L_{j1}^{dual})^{+} \le \\ &L_{j2}^{cost} - L_{j2}^{max} \cdot (L_{j2}^{dual})^{+} \le \\ &L_{j2}^{cost} - \min\{L_{j2}^{max}, Q - L_{j2}^{load} - d_k\} \cdot (L_{j2}^{dual})^{+} \end{aligned} \tag{5.86}$$

Subcase 3-2: $L_{j1}^{max} \geq Q - L_{j1}^{load} - d_k$ and $L_{j2}^{max} \leq Q - L_{j2}^{load} - d_k$. In this case, Inequality (5.84) can be rewritten as

$$L_{j1}^{cost} - (Q - L_{j1}^{load} - d_k) \cdot (L_{j1}^{dual})^+ \leq L_{j2}^{cost} - L_{j2}^{max} \cdot (L_{j2}^{dual})^+. \tag{5.87}$$

The subcase implies that

$$L_{j2}^{load} - L_{j1}^{load} + L_{j2}^{max} \leq Q - d_k - L_{j1}^{load}.$$

By this inequality and criterion 7 of Proposition 8, we obtain exactly the required Inequality (5.87) as

$$\begin{aligned}
&L_{j1}^{cost} - (Q - d_k - L_{j1}^{load}) \cdot (L_{j1}^{dual})^+ \leq \\
&L_{j1}^{cost} - (L_{j2}^{load} - L_{j1}^{load} + L_{j2}^{max}) \cdot (L_{j1}^{dual})^+ \leq \\
&L_{j2}^{cost} - L_{j2}^{max} \cdot (L_{j2}^{dual})^+
\end{aligned} \tag{5.88}$$

Subcase 3-3: $L_{j1}^{max} \geq Q - L_{j1}^{load} - d_k$ and $L_{j2}^{max} \geq Q - L_{j2}^{load} - d_k$. Then, the inequality to be established based on Inequality (5.84) becomes

$$L_{j1}^{cost} - (Q - L_{j1}^{load} - d_k) \cdot (L_{j1}^{dual})^+ \leq L_{j2}^{cost} - (Q - L_{j2}^{load} - d_k) \cdot (L_{j2}^{dual})^+. \tag{5.89}$$

Consider criterion 7 in the statement of Proposition 8, and rewrite it in the form

$$L_{j1}^{cost} - (L_{j2}^{load} - L_{j1}^{load}) \cdot (L_{j1}^{dual})^+ \leq L_{j2}^{cost} + L_{j2}^{max} \cdot ((L_{j1}^{dual})^+ - (L_{j2}^{dual})^+).$$

Together with criterion 6 of Proposition 8, this inequality implies that the following inequality holds for all values of q in the interval $[0, L_{j2}^{max}]$ (because it holds at its endpoints):

$$L_{j1}^{cost} - (L_{j2}^{load} - L_{j1}^{load}) \cdot (L_{j1}^{dual})^+ \leq L_{j2}^{cost} + q \cdot ((L_{j1}^{dual})^+ - (L_{j2}^{dual})^+).$$

Consider now the inequality when $q = Q - L_{j2}^{load} - d_k$, which is in $[0, L_{j2}^{max}]$ by hypothesis. It becomes

$$L_{j1}^{cost} - (L_{j2}^{load} - L_{j1}^{load}) \cdot (L_{j1}^{dual})^+ \leq L_{j2}^{cost} + (Q - L_{j2}^{load} - d_k) \cdot ((L_{j1}^{dual})^+ - (L_{j2}^{dual})^+),$$

which is equivalent to (5.89) after some simplifications. This concludes the proof.

□

5.5 Computational experiments

Future work will be required to assess the computational performance of the approaches described in this chapter. Numerical results can be compared with a similar model in which only full delivery shifts are allowed.

5.6 Conclusions

We have presented two MILP formulations of a two-period VRP with partial delivery shifts, where a linear penalty is incurred proportional to the quantity shifted. We have developed a column-row generation algorithm to solve the LP-relaxation of the first formulation. Columns correspond to decision variables on routes and delivery quantities, whereas new rows are generated to guarantee that the vehicle capacity is respected in the new routes. We have also developed a column generation algorithm to solve the LP-relaxation of the second formulation, where only columns with extreme delivery patterns are introduced to the master problem. All other feasible delivery patterns are obtained as a convex combination of the extreme delivery patterns in the master problem. While the two MILP formulations are equivalent, their LP-relaxations are not. By using a small example, we have shown that the LP-relaxation of the second MILP is tighter than the first one. However, without computational experiments it is hard to say which of the two MILP formulations can be solved more efficiently in terms of the computation time. On the one hand, the column-row generation algorithm developed to solve the LP-relaxation of the first MILP adds a new constraint to the master problem for each new route, whereas the column generation algorithm developed to solve the LP-relaxation of the second MILP does not. On the other hand, the label-setting algorithm developed to solve the pricing problem of the column-row generation algorithm appears to create much less labels as compared to the one in the column generation.

Chapter 6

Conclusions and future studies

The topic examined in this thesis has been the IRP for perishables, which is a common decision making problem in inventory control and distribution of fresh products in food retail chains. The main decisions are: (a) how often each store should be served, (b) how much should be delivered to each store, and (c) how the stores should be incorporated into delivery routes. While retail managers are facing the same problem on daily basis, they do not follow a unique replenishment and distribution policy. This research has postulated three problems to deal with a simplified version of the real problem by emphasizing on the synchronization between replenishment and distribution decisions.

Chapter 2 has discussed several relevant topics to this thesis, namely, the VRP, MPVRP, PVRP, and IRP. Moreover, we have reviewed inventory control of perishables in RMI and VMI systems. We have stated the underlying assumptions, the main parameters, the decision variables, and the objective function for each problem type.

In Chapter 3, an SIRP for a single perishable product has been investigated. Profit maximization is the main objective in this problem, while a high customer service level is imposed as a side constraint and freshness is regarded as a consequence of optimizing the profit. We have developed and compared variant solution methods to solve the SIRP for perishables, and we have analyzed the results. We have shown that by considering uncertainty and combining inventory with routing decisions for perishable products, retail chains can observe a significant increase in their net profit. We have also shown how such benefit can be gained and quantified. Moreover, we have measured the value of considering uncertainty and the value of accessing full information on future demands. Our numerical results show that a simple deliver up-to-level policy performs almost as efficiently as other more complicated methods when the target service level is high. In the most sophisticated and yet efficient solution method developed for the SIRP, i.e., the decomposition-integration (*DI*) method, we come to an optimization problem for which a Matheuristic algorithm has been proposed. However, the optimization problem can also be solved to optimality given some assumptions. Chapters 4 and 5 discuss such assumptions and the exact solution methods.

In Chapter 4, we have introduced a two-period VRP where orders of each period can be shifted to the other period and change in quantity. Full delivery shifts, as compared to partial delivery shifts, is the underlying assumption to solve the two-period VRP. Although this problem is emanated from the *DI* solution method to solve the SIRP, we bring it up as an independent problem. An efficient branch-and-price algorithm based on classical and new problem-specific acceleration techniques has been implemented to solve this model. Even though the two-period VRP turns out to be considerably harder than the classical VRP, our algorithm yields provably good solutions for many instances of the problem in a reasonable time. The experiments demonstrate that, compared to solving two independent VRPs, the routing costs and the number of vehicles can decrease significantly when orders are allowed to be shifted. This implies that there is potential value in handling the two-period VRP model, as opposed to solving two independent VRP models. The results also suggest that, if one wants to avoid the computational burden of solving large two-period VRPs to optimality, identifying unpromising shifts may reduce the size of the instances to be solved while still producing economies in transportation costs.

In Chapter 5, the same two-period VRP is considered where the orders placed by stores for each period can be partially shifted to the other period, given that the sum of the delivery quantities in two periods to each customer is a fix value. The shifts are at the cost of a penalty linearly proportional to the quantity shifted. We have represented two MILP formulations for the problem. Moreover, we have demonstrated that these two formulations are equivalent. A column-row generation algorithm to solve the LP-relaxation of the first formulation has been developed. For the second formulation, we have developed a column generation algorithm to solve it. Details of two label-setting algorithms have been discussed; a label-setting algorithm to solve the pricing problem raised in the column-row generation algorithm and another to solve the pricing problem of the column generation algorithm. Numerical results can be compared with the model with full delivery shifts.

Each of the three aforementioned problems is imposing some simplification assumptions on the real IRP for perishables in food retail chains. The biggest obstacle of implementing the first model in practice is that it considers a single product, whereas fresh products are carried together in refrigerated vehicles. Therefore, extending the models and adapting the solution methods to include multiple perishable products is the most promising future research direction. However, this is not a barrier for the two-period VRPs with full or partial delivery shifts; the models and the solution methods are conveniently extendable to include multiple products. Another real assumption to be considered is to let multiple routes for each vehicle, and multiple visits to each store in each period, which can be regarded as another future research to approach the real problem.

Applying different policies for the selling price in the SIRP for perishables in Chapter 3 sounds an appealing extension of the work. Considering a and p as acquisition and selling price of each unit of a perishable product with maximum shelf life of L, we have tested two discounting policies; (1) a linearly decreasing selling price from s to a during the shelf life and (2) selling price of s during the first $L-1$ periods and a in the last period of shelf life. Imposing either of these policies leads to every day visits to all stores as the best delivery frequency. This is due to the fact that in food retail chains, transportation cost is less than 7% of the revenue, compared to acquisition cost which is almost 60% of the revenue. However, considering lower discounts may result in less frequent deliveries, and this can be examined in future studies.

Bibliography

Abdelmaguid T F, Dessouky M M (2006) A genetic algorithm approach to the integrated inventory-distribution problem. *International Journal of Production Research*, 44(21):4445-4464.

Adachi Y, Nose T, Kuriyama S (1999) Optimal inventory control policy subject to different selling prices of perishable commodities. *International Journal of Production Economics*, 60:389-394.

Adelman D (2004) A price-directed approach to stochastic inventory routing. *Operations Research*, 52(4):499-514.

Adulyasak Y, Cordeau J F, Jans R (2014) Formulations and branch-and-cut algorithms for multi-vehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26(1):103-120.

Albareda-Sambola M, Fernandez E, Laporte G (2014) The dynamic multiperiod vehicle routing problem with probabilistic information. *Computers & Operations Research*, 48:31-39.

Alegre J, Laguna M, Pacheco J (2007) Optimizing the periodic pickup of raw materials for a manufacturer of auto parts. *European Journal of Operational Research*, 179(3):736-746.

Al Shamsi A, Al Raisi A, Aftab M (2014) Pullution inventory routing problem with perishable goods. Golinska P, ed. *Logistics Operations*, Supply Chain Management and Sustainability, EcoProduction, Springer, Switzerland, 585-596.

Andersson H, Hoff A, Christiansen M, Hasle G, Lokketangen A (2010) Industrial aspects and literature survey: combined inventory management and routing. *Computers & Operations Research*, 37:1515-1536.

Andreatta G, Lulli G (2008) A multi-period TSP with stochastic regular and urgent demands. *European Journal of Operational Research*, 185:122-132.

Angelelli E, Savelsbergh M, Speranza M G (2007a) Competitive analysis for dynamic multi-period uncapacitated routing problems. *Networks*, 49(4):318-329.

Angelelli E, Savelsbergh M, Speranza M G (2007b) Competitive analysis of a dispatch policy for a dynamic multi-period routing problem. *Operations Research Letters*, 35:713-721.

Angelelli E, Bianchessi N, Mansini R, Speranza M G (2009) Short term strategies for a dynamic multi-period routing problem. *Transportation Research Part C*, 17:106-119.

Archetti C, Bertazzi L, Hertz A, Speranza M G (2012) A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24:101-116.

Archetti C, Bertazzi L, Laporte G, Speranza M G (2007) A branch-and-cut algorithm for a vendor managed inventory routing problem. *Transportation Science*, 41:382-391.

Archetti C, Bouchard M, Desaulniers G (2011) Enhanced branch and price and cut for vehicle routing with split deliveries and time windows. *Transportation Science*, 45(3):285-298.

Archetti C, Jabali O, Speranza M G (2015) Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 61:122-134.

Archetti C, Savelsbergh M, Speranza M G (2006) Worst-case analysis for split delivery vehicle routing problems. *Transportation Science*, 40(2):226-234.

Archetti C, Speranza M G (2014) A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, Published online.

Archetti C, Speranza M G (2016) The inventory routing problem: the value of integration. *International Transactions in Operational Research*, 23(3):393-407.

Archetti C, Speranza M G, Hertz A (2006) A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, 40(1):64-73.

Athanasopoulos T, Minis I (2013) Efficient techniques for the multi-period vehicle routing problem with time windows within a branch and price framework. *Annals of Operations Research*, 206:1-22.

Azi N, Gendreau M, Potvin J Y (2010) An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, 202(3):756-763.

Baldacci R, Bartolini E, Mingozzi A, Valletta A (2011) An exact algorithm for the period routing problem. *Operations Research*, 59:228-241.

Baldacci R, Mingozzi A, Roberti R (2011) New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269-1283.

Bard J F, Nananukul N (2010) A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, 37:2202-2217.

Barnes-Schuster D, Bassok Y (1997) Direct shipping and the dynamic single-depot multi-retailer inventory system. *European Journal of Operational Research*, 101(3):509-518.

Barnhart C, Johnson E L, Nemhauser G L, Savelsbergh M, Vance P H (1996) Branch-and-price: column generation for solving huge integers programs. *Operations Research*, 46(3):316-328.

Beltrami E J, Bodin L D (1974) Networks and vehicle routing for municipal waste collection. *Networks*, 4:65-94.

Berk E, Gurler U (2008) Analysis of the (Q,r) inventory model for perishables with positive lead times and lost sales. *Operations Research*, 56(5):1238-1246.

Berman O, Larson R C (2001) Deliveries in an inventory routing problem using stochastic dynamic programming. *Transportation Science*, 35(2):192-213.

Bertazzi L (2008) Analysis of direct shipping policies in an inventory routing problem with discrete shipping times. *Management Science*, 54(4):748-762.

Bertazzi L, Paletta G, Speranza M G (2002) Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36:119-132.

Bertazzi L, Speranza M G (2011) Matheuristics for inventory routing problems. In: Montoya-Torres J R, Juan A A, Huatuco L H, Faulin J, Rodriguez-Verjan G L. *Hybric algorithms for service, computing, and manufacturing systems: routing and scheduling solutions*, IGI Global, Hershey, 1-14.

Bertazzi L, Speranza M G (2012) Inventory routing problem: an introduction. *EURO Journal on Transportation and Logistics*, 1:307-326.

Bertazzi L, Speranza M G (2013) Inventory routing problem with multiple customers. *EURO Journal on Transportation and Logistics*, 2:255-275.

Bilgen B, Celebi Y (2013) Integrated production scheduling and distribution planning in dairy supply chain by hybrid modelling. *Annals of Operations Research*, 211:55-82.

Broekmeulen R, Van Donselaar K (2009) A heuristic to manage perishable inventory with batch ordering, positive lead times, and time-varying demand. *Computers & Operations Research*, 36:3013-3018.

Burnetas A N, Smith C E (2000) Adaptive ordering and pricing for perishable products. *Operations Research*, 48(3):436-443.

Campbell A M, Clarke L, Kleywegt A J, Savelsbergh M W P (1998) The inventory routing problem. Crainic T G, Laporte G, eds. *Fleet Management and Logistics*, Springer, Boston, 95-113.

Campbell A M, Savelsbergh M (2004) Delivery volume optimization. *Transportation Science*, 38(2):210-223.

Campbell A M, Savelsbergh M (2004) A decomposition approach for the inventory routing problem. *Transportation Science*, 38(4):488-502.

Campbell A M, Wilson J H (2014) Forty years of periodic vehicle routing. *Networks*, 63(1):2-15.

Chan M L A, Federgruen A, Simchi-Levi D (1998) Probabilistic analyses and practical algorithms for inventory routing models. *Operations Research*, 46(1):96-106.

Chao X, Gong X, Shi C, Zhang H (2015) Approximation algorithms for perishable inventory systems. *Operations Research*, 63(3):585-601.

Chen X, Pang Z, Pan L (2014) Coordinating inventory control and pricing strategies for perishable products. *Operations Research*, 62(2):284-300.

Chen L M, Sapra A (2013) Joint inventory and pricing decisions for perishable products with two-period lifetime. *Naval Research Logistics*, 60(5):343-366.

Chiu H N (1995) A heuristic (R,T) periodic review perishable inventory model with lead times. *International Journal of Production Economics*, 42:1-15.

Christiansen M, Fagerholt K, Flatberg T, Haugen Ø, Kloster O, Lund EH (2011) Maritime inventory routing with multiple products: a case study from the cement industry. *European Journal of Operational Research*, 208(1):86-94.

Christofides N, Beasley J E (1984) The period routing problem. *Networks*, 14:237-256.

Clarke G, Wright J W (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568-581.

Coelho L C, Cordeau J F, Laporte G (2012) Consistency in multi-vehicle inventory routing. *Transportation Research Part C: Emerging Technologies*, 24:270-287.

Coelho L C, Cordeau J F, Laporte G (2012) The inventory routing problem with transshipment. *Computers & Operations Research*, 39:2537-2548.

Coelho L C, Cordeau J F, Laporte G (2012) Dynamic and stochastic inventory routing. *CIRRELT-2012-37*, 1-27.

Coelho L C, Cordeau J F, Laporte G (2014) Thirty years of inventory routing. *Transportation Science*, 48(1):1-19.

Coelho L C, Cordeau J F, Laporte G (2014) Heuristics for dynamic and stochastic inventory routing. *Computers & Operations Research*, 52:55-67.

Coelho L C, Laporte G (2013) A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research*, 51:7156-7169.

Coelho L C, Laporte G (2013) The exact solution of several classes of inventory routing problems. *Computers & Operations Research*, 40:558-565.

Coelho L C, Laporte G (2014) Optimal joint replenishment, delivery and inventory management policies for perishable products. *Computers & Operations Research*, 47:42-52.

Coelho L C, Laporte G (2014) Improved solutions for inventory routing problems through valid inequalities and input ordering. *International Journal of Production Economics*, 155:391-397.

Cordeau J F, Gendreau M, Laporte G (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30: 105-119.

Cooper W L (2001) Pathwise properties and performance bounds for a perishable inventory system. *Operations Research*, 49(3):455-466.

Dabia S, Ropke S, Van Woensel T, De Kok T (2013) Branch and price for the time-dependent vehicle routing problem with time windows. *Transportation Science*, 47(3):380-396.

Dell'Amico M, Righini G, Salani M (2006) A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science*, 40(2):235-247.

Deniz B, Karaesmen I, Scheller-Wolf A (2010) Managing perishables with substitution: Inventory issuance and replenishment heuristics. *Manufacturing & Service Operations Management*, 12(2):319-329.

Desaulniers G (2010) Branch-and-price-and-cut for the split delivery vehicle routing problem with time windows. *Operations Research*, 58(1):179-192.

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

Du Merle O, Villeneuve D, Desrosiers J, Hansen P (1999) Stabilized column generation. *Discrete Mathematics*, 194:229-237.

Ekici A, Ozener O O, Kuyzu G (2015) Cyclic delivery schedules for an inventory routing problem. *Transportation Science*, 49(4):817-829.

Federgruen A, Prastacos G, Zipkin PH (1986) An allocation and distribution model for perishable products. *Operations Research*, 34(1):75-82.

Federgruen A, Zipkin P (1984) A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019-1037.

Feillet D (2010) A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR*, 8(4):407-424.

Feillet D, Dejax P, Gendreau M, Gueguen C (2004) An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. *Networks*, 44(3):216-229.

Francis P, Smilowitz K, Tzur M (2006) The period vehicle routing problem with service choice. *Transportation Science*, 40:439-454.

Francis P, Smilowitz K, Tzur M (2008) The period vehicle routing problem and its extensions. *The vehicle routing problem: latest advances and new challenges*, Golden B, Raghavan S, and Wasil E (Editors), Springer, Berlin, 73-102.

Gallego G, Simchi-Levi D (1990) On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer R-systems. *Management Science*, 36:240-243.

Gaudioso M, Paletta G (1992) A heuristic for the periodic vehicle routing problem. *Transportation Science*, 26:86-92.

Gaur V, Fisher ML (2004) A periodic inventory routing problem at a supermarket chain. *Operations Research*, 52(6):813-822.

Goyal S K, Giri B C (2001) Recent trends in modeling of deteriorating inventory. *European Journal of Operational Research*, 134:1-16.

Gronhaug R, Christiansen M, Desaulniers G, Desrosiers J (2010) A branch-and-price method for a liquified natural gas inventory routing problem. *Transportation Science*, 44(3):400-415.

Gutierrez-Jarpa G, Desaulniers G, Laporte G, Marianov V (2010) A branch-and-price algorithm for the vehicle routing problem with deliveries, selective pickups and time windows. *European Journal of Operational Research*, 206:341-349.

Hemmelmayr V, Doerner K F, Hartl R F (2009) A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195:791-802.

Hemmelmayr V, Doerner K F, Hartl R F, Savelsbergh M W P (2009) Delivery strategies for blood products supplies. *OR Spectrum*, 31:707-725.

Hemmelmayr V, Doerner K F, Hartl R F, Savelsbergh M W P (2010) Vendor managed inventory for environments with stochastic product usage. *European Journal of Operational Research*, 202:686-695.

Hsieh T P, Dye C Y (2010) Optimal replenishment policy for perishable items with stock-dependent selling rate and capacity constraint. *Computers & Industrial Engineering*, 59:251-258.

Hsu V N (2000) Dynamic economic lot size model with perishable inventory. *Management Science*, 46(8):1159-1169.

Huang S H, Lin P C (2010) A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. *Transportation Research Part E*, 46:598-611.

Irnich S, Desaulniers G (2005) Shortest path problems with resource constraints. Desaulniers G, Desrosiers J, Solomon M M, eds. *Column Generation* (Springer, New York), 33-65.

Jaillet P, Bard J F, Huang L, Dror M (2002) Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science*, 36(3):292-300.

Jepsen M, Petersen B, Spoorendonk S, Pisinger D (2008) Subset-row inequalities applied to the vehicle routing problem with time windows. *Operations Research*, 56(2):497-511.

Karaesmen I Z, Scheller-Wolf A, Deniz B (2011) Managing perishable and aging inventories: Review and future research directions. Kempf K G, Keskinocak P, Uzsoy R, eds. *Planning Production and Inventories in the Extended Enterprise*, International Series in Operations Research and Management Science. Springer, New York, 151:393-436.

Kleywegt A J, Nori V S, Savelsbergh M W P (2002) The stochastic inventory routing problem with direct deliveries. *Transportation Science*, 36(1):94-118.

Kleywegt A J, Nori V S, Savelsbergh M W P (2004) Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science*, 38(1):42-70.

Kouki C, Jemai Z, Minner S (2015) A lost sales (r,Q) inventory control model for perishables with fixed lifetime and lead time. *International Journal of Production Economics*, 168:143-157.

Kouki C, Jouini O (2015) On the effect of lifetime variability on the performance of inventory systems. *International Journal of Production Economics*, 167:23-34.

Kouki C, Sahin E, Jemai Z, Dallery Y (2014) Analysis of a periodic review inventory control system with perishables having random lifetime. *International Journal of Production Research*, 52(1):283-298.

Laporte G (2009) Fifty years of vehicle routing. *Transportation Science*, 43(4):408-416.

Laporte G, Nobert Y, Desrochers M (1985) Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5):1050-1073.

Le T, Diabat A, Richard J P, Yih Y (2013) A column generation-based heuristic algorithm for an inventory routing problem with perishable goods. *Optimization Letters*, 7:1481-1502.

Li Y, Lim A, Rodrigues B (2009) Pricing and inventory control for a perishable product. *Manufacturing & Service Operations Management*, 11(3):538-542.

Lian Z, Liu L (1999) A discrete-time model for perishable inventory systems. *Annals of Operations Research*, 87:103-116.

Lübbecke M E, Desrosiers J (2005) Selected topics in column generation. *Operations Research*, 53(6):1007-1023.

Mancini S (2016) A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: formulation and adaptive large neighborhood search based matheuristic. *Transportation Research Part C*, 70:100-112.

Matos A C, Oliveira R C (2004) An experimental study of the ant colony system for the period vehicle routing problem. *Ant Colony Optimization and Swarm Intelligence, Vol. 3172 of Lecture Notes in Computer Science*, Dorigo M, Birattari M, Blum C, Gambardella L M, Mondada F, and Stutzle T (Editors), Springer, Berlin, 286-293.

Minkoff A S (1993) A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. *Operations Research*, 41(1):77-90.

Minner S, Transchel S (2010) Periodic review inventory control for perishable products under service level constraints. *OR Spectrum*, 32:979-996.

Mirzaei S, Seifi A (2015) Considering lost sale in inventory routing problems for perishable goods. *Computers & Industrial Engineering*, 87:213-227.

Mourgaya M, Vanderbeck F (2007) Column generation based heuristic for tactical planning in multi-period vehicle routing. *European Journal of Operational Research*, 183:1028-1041.

Nahmias S (2011) Perishable inventory systems. Hillier F S, ed. *International Series in Operations Research and Management*, (Springer, NewYork), 1-80.

Nandakumar P, Morton T E (1993) Near myopic heuristics for the fixed-life perishability problem. *Management Science*, 39(12):1490-1498.

Olsson F, Tydesjo P (2010) Inventory problems with perishable items: Fixed lifetimes and backlogging. *European Journal of Operational Research*, 202:131-137.

Ozener O, Ergun O, Savelsbergh M (2013) Allocation cost of service to customers in inventory routing. *Operations Research*, 60(5):1-14.

Pillac V, Gendreau M, Gueret C, Medaglia A L (2013) A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225:1-11.

Pirkwieser S, Raidl G R (2010), Multi-level variable neighborhood search for periodic routing problems. *Evolutionary Computation in Combinatorial Optimization, Vol. 6022 of Lecture Notes in Computer Science*, Cowling P and Merz P (Editors), Springer, Berlin, 226-238.

Popović D, Vidović M, Radivojević G (2012) Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. *Expert Systems Applications*, 39(18):13390-13398.

Powell W E (2011) *Approximate dynamic programming: solving the curses of dimensionality*, (Wiley, New Jersey).

Puterman M L (1994) *Markov decision processes*. John Wiley & Sons, New York.

Qu W W, Bookbinder J H, Iyogun P (1999) An integrated inventory transportation system with modified periodic policy for multiple products. *European Journal of Operational Research*, 115(2):254-269.

Raa B, Aghezaaf E H (2008) Designing distribution patterns for long-term inventory routing with constant demand rates. *International Journal of Production Economics*, 112:255-263.

Raa B, Aghezaaf E H (2009) A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research*, 192:429-441.

Reinman M I, Rubio R, Wein L M (1999) Heavy traffic analysis of the dynamic stochastic inventory routing problem. *Transportation Science*, 33(4):361-380.

Righini G, Salani M (2006) Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3:255-273.

Rusdiansyah A, Tsao D B (2005) An integrated model of the periodic delivery problems for vending machine supply chains. Journal of Food Engineering, 70:421-434.

Russell R, Igo W (1979) An assignment routing problem. *Networks*, 9:1-17.

Schwartz LB, Ward JE, Zhai X (2006) On the interactions between routing and inventory management policies in a one-warehouse N-retailer distribution system. *Manufacturing & Service Operations Management*, 8:253-272.

Solomon M M (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254-265.

Solyali O, Süral H (2011) A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory routing problem. *Transportation Science*, 45(3):335-345.

Soysal M, Bloemhof-Ruwaard J M, Haijema R, Van der Vorst J (2015) Modeling an inventory routing problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics*, 164:118-133.

Stacey J, Natarajarathinam M, Sox C (2007) The storage constrained inbound inventory routing problem. *International Journal of Physical Distribution & Logistics Management*, 37(6):484-500.

Van Anholt R G, Coelho L C, Laporte G, Vis I F A (2016) An inventory routing problem with pickups and deliveries arising in the replenishment of automated teller machines. *Transportation Science*, *http://dx.doi.org/10.1287/trsc.2015.0637*.

Van Donselaar K H, Gaur V, Van Woensel T, Broekmeulen R, Fransoo J C (2010) Ordering behavior in retail stores and implications for automated replenishment. *Management Science*, 56(5):766-784.

Van Donselaar K, Van Woensel T, Broekmeulen R, Fransoo J (2006) Inventory control of perishables in supermarkets. *International Journal of Production Economics*, 104:462-472.

Vidal T, Crainic T G, Gendreau M, Lahrichi N, Rei W (2012) A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems. *Operations Research*, 60:611-624.

Wen M, Cordeau J F, Laporte G, Larsen J (2010) The dynamic multi-period vehicle routing problem. *Computers & Operations Research*, 37:1615-1623.

Yu Y, Chu C, Chen H, Chu F (2012) Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research*, 197(1):135-158.

Euro Bank (2009) European Bank for Reconstruction and Development and Food and Agriculture Organization of the United Nations, Food Retail Handbook, *http://www.fao.org/fileadmin/user_upload/tci/docs/AH4-Food%20Retail.pdf*.

FMI (2014) Food Marketing Industry, Supermarket Facts, *http://www.fmi.org/research-resources/supermarket-facts*.

NAICS (2012) Canadian industry statistics on grocery stores (NAICS 4451), *https://www.ic.gc.ca/app/scr/sbms/sbb/cis/revenues.html?code=4451&lang=eng*.