Optimization models and solution methods for inventory routing problems

Aldair Alberto Álvarez Díaz

São Carlos
March, 2020

Optimization models and solution methods for inventory routing problems [1]

Aldair Alberto Álvarez Díaz

Supervisors: Prof. Dr. Pedro Munari
Prof. Dr. Reinaldo Morabito

Doctoral dissertation submitted in fulfillment of the requirements for the degree of Doctor in Production Engineering at the Federal University of São Carlos.

São Carlos
March, 2020

[1]This research was supported by the São Paulo Research Foundation (FAPESP), grants 2017/06664-9 and 2017/13739-5; the National Council for Scientific and Technological Development (CNPq), grant 153046/2016-3; and the Coordination for the Improvement of Higher Education Personnel (CAPES)

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Aldair Alberto Álvarez Diaz, realizada em 13/03/2020:

Prof. Dr. Pedro Augusto Munari Junior
UFSCar

Prof. Dr. Reinaldo Morabito Neto
UFSCar

Prof. Dr. Roberto Fernandes Tavares Neto
UFSCar

Prof. Dr. Maristela Oliveira dos Santos
USP

Prof. Dr. Rafael Martinelli Pinto
PUC/RJ

Prof. Dr. Silvio Alexandre de Araujo
UNESP

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Prof. Dr. Pedro Augusto Munari Junior

Acknowledgments

The past four years have been an exciting journey that I have not made by myself. I have always had the support of an all-star team to which I am grateful for helping me reach this stage. These pages are devoted to acknowledge their contributions and to express my sincere gratitude to them.

First of all, I thank my supervisors, Pedro Munari and Reinaldo Morabito, for being the leaders of this team, for mentoring me throughout these years and always making space for me in their busy schedules. The time and effort they committed to me and my research certainly made a huge difference in the outcomes of my PhD. Professor Pedro, I will always be grateful for being more than a supervisor to me since the beginning of this academic partnership. Your guidance and advice, as well as your concern about my well-being, made my life easier all this time in Brazil. Your encouragement made me grow as a researcher and as a person, expanding my limits. Professor Reinaldo, your experience and insights have enriched the way I see the research. I feel honored for having such a bright, professional and, above all, kind person as one of my supervisors. I thank you both for allowing me to do (more than) research together.

I am also thankful to Jean-François Cordeau and Raf Jans for their supervision during my stay in Montreal. My time there was a unique experience, full of learning and enrichment in both scientific and personal terms. I could not have had such an amazing experience without your professionalism and competence to guide my research, as a complement to the incredible atmosphere that the city provides.

My scientific achievements would not have been possible without a proper working environment in every single stage of my PhD. I therefore thank my friends and colleagues at the GPO-UFSCar and CIRRELT for supporting me and for the many good times we shared. I also thank the secretaries of the PPGEP-UFSCar and the university staff for their behind-the-scenes work that allowed me to successfully complete this journey.

No acknowledgment of mine would be complete without thanking my friends, for always having my back, sticking to me all the time, encouraging me to keep going, and for being my cheerleaders even from a distance. In particular, I thank *Yeyo* and *Luchy*, my dearest lifelong friends. I am also grateful to Karen Montes, whose company, support and patience for so many years were an invaluable contribution to this thesis and a permanent gift for my personal development.

Por último, pero no menos importante, agradezco a mi familia por el amor y apoyo incodicionales que siempre me han brindado. A mis padres, Delcy y Julio, a quienes debo todo lo que soy y tengo hoy en día, les agradezco por incentivarme a seguir adelante y perseguir mis metas. A mis amados hermanos, Karen e Julio Jr., por quienes me esfuerzo para ser el mejor ejemplo posible, les agradezco su apoyo y les dedico especialmente este trabajo. A ustedes, mi familia, siempre los llevo presentes a pesar de la distancia, pues son mi mayor regalo y motivo para perseverar.

Finally, I am grateful for the financial support of the São Paulo Research Foundation (FAPESP), grants 2017/06664-9 and 2017/13739-5; the National Council for Scientific and Technological Development (CNPq), grant 153046/2016-3; and the Coordination for the Improvement of Higher Education Personnel (CAPES).

This thesis shows not only the scientific results of my research but also how teamwork helped me to achieve my goals successfully. Throughout these years I have gained a lot, completely changing the course of my life. This has been, without a doubt, the most rewarding journey of my life :)

Aldair

Abstract

Inventory management and distribution planning are essential activities for an efficient performance in the supply chain, especially for companies operating under the vendor-managed inventory business model. In this model, suppliers are allowed to manage the inventory levels and purchasing orders of their customers with the aim of reducing logistics and improving the supply chain performance. When inventory management and distribution planning are addressed in an integrated way in the vendor-managed inventory context, a challenging optimization problem arises, the inventory routing problem (IRP). In the IRP, a supplier is responsible for simultaneously determining the replenishment plan for its customers throughout a planning horizon as well as the vehicle routing and scheduling plan in each period such that a given performance measure is optimized.

The integrated optimization of inventory management and distribution planning activities can provide significant competitive advantages for companies. However, despite its practical appeal and benefits, the IRP has received increasing attention only in the last years. Consequently, there is still a considerable lack of research regarding optimization models and specific solution methods for relevant practical variants of this problem. Thus, the objective of this thesis is to develop comprehensive mathematical models and effective solution methods for several IRPs. Relevant variants are considered to make the addressed problems as realistic as possible.

Firstly, we describe the basic variant of the IRP and present a mathematical formulation for this problem. We then present two metaheuristic algorithms based on iterated local search and simulated annealing to solve this variant. Two different objective functions are considered. The results of extensive computational experiments using problem instances from the literature show that the presented metaheuristic algorithms effectively handle both objective functions, providing high-quality solutions within relatively short running times. In addition, the metaheuristics were able to find new best solutions for some of the benchmark instances.

Then we shift to a practical variant of the IRP considering product perishability. This feature has a substantial relevance in the supply chain context given that in several industries, the raw materials, as well as intermediate and final products, are often perishable. Moreover, perishability may appear in more than one activity throughout the supply chain. We study a variant in which the product is assumed to have a fixed shelf-life with age-dependent revenues and inventory holding costs. We first introduce four different mathematical formulations and branch-and-cut algorithms to solve them. We also propose a hybrid heuristic based on the combination of an iterated local search metaheuristic and two mathematical programming components. The results of computational experiments show the different advantages of the introduced formulations and the effectiveness of our hybrid method when dealing with this variant as well as the basic variant of the problem.

Finally, we focus on a stochastic variant of the IRP. Uncertainty plays a crucial role in supply chain management given that critical input data that are required for effective planning often are not known in advance. We address the basic variant of the IRP under the consideration that both the product supply and the customer demands are uncertain. We introduce a two-stage stochastic programming formulation and a heuristic solution method for this problem. From the results of extensive computational experiments, we show the response mechanisms of the optimal solutions under different uncertainty levels and cost configurations. We also show that the heuristic method effectively solves instances with a large number of scenarios.

By investigating different practical constraints for the IRP and providing tailored effective solution methods for the studied variants, this thesis addresses problems arising in several logistics contexts and shows the adaptability of the basic variant of the IRP and how it can be used as a basis to study richer practical IRPs. It brings contributions for the supply chain optimization literature and for the development of tools for supporting decision-making in practice.

Keywords: inventory routing; branch-and-cut; metaheuristics; hybrid methods; product perishability; stochastic programming.

Resumo

A gestão de estoques e o planejamento da distribuição são atividades essenciais para um desempenho eficiente na cadeia de suprimentos, especialmente para empresas que operam sob o modelo de estoque gerenciado pelo fornecedor. Nesse modelo, os fornecedores podem gerenciar os níveis de estoque e as ordens de compra de seus próprios clientes, com o objetivo de reduzir custos logísticos e melhorar o desempenho da cadeia de suprimentos. Quando a gestão de estoques e o planejamento da distribuição são tratados de forma integrada aparece um problema de otimização desafiador, conhecido como o problema de roteamento de estoques (PRE). No PRE, um fornecedor deve determinar simultaneamente o plano de reabastecimento para seus clientes em um horizonte de planejamento e a programação das rotas de entrega em cada período de forma que uma determinada medida de desempenho seja otimizada.

A otimização integrada das atividades da gestão de estoques e do planejamento da distribuição pode fornecer vantagens competitivas para as empresas. No entanto, apesar de seu apelo prático e dos benefícios substanciais que essa otimização pode fornecer, o PRE recebeu uma atenção crescente apenas nos últimos anos. Portanto, ainda existe uma considerável falta de pesquisa no que tange a métodos de solução específicos para variantes práticas relevantes desse problema. Assim, o objetivo dessa tese é desenvolver modelos matemáticos abrangentes e métodos de solução eficazes para diversos PREs. Variantes práticas são consideradas para tornar os problemas abordados o mais realista possível.

Em primeiro lugar, descreve-se a variante básica do PRE e apresenta-se uma formulação matemática para esse problema. Dois algoritmos metaheurísticos, baseados em busca local iterada e *simulated annealing*, são apresentados para resolver a variante básica do PRE, considerando duas funções objetivo diferentes. Os resultados de experimentos computacionais usando instâncias da literatura mostram que os dois algoritmos metaheurísticos podem fornecer soluções de alta qualidade em tempos relativamente curtos para ambas as funções objetivo. Além disso, as metaheurísticas conseguiram encontrar novas melhores soluções para algumas dessas instâncias.

Em seguida, estuda-se uma variante prática do PRE considerando a perecibilidade do produto. A perecibilidade tem uma relevância significativa no contexto da cadeia de suprimentos dado que, em muitas indústrias, as matérias-primas bem como os produtos intermediários e finais são perecíveis. Além disso, a perecibilidade pode aparecer em mais de uma atividade em toda a cadeia de suprimentos. Na variante estudada, supõe-se que o produto tem uma vida útil pré-definida além de receitas e custos de estocagem dependentes da idade do produto. Para essa variante, apresenta-se quatro formulações matemáticas e algoritmos do tipo *branch-and-cut* para resolvê-las. Além disso, apresenta-se uma heurística híbrida baseada na combinação de uma metaheurística de busca local iterada e dois componentes de programação matemática. Os resultados de experimentos computacionais mostram as diferentes vantagens das formulações apresentadas e a capacidade do método híbrido para lidar com essa variante, assim como com a variante básica do problema.

Finalmente, uma variante estocástica do PRE é abordada. As incertezas desempenham um papel crucial na gestão da cadeia de suprimentos, dado que informações críticas necessárias para um planejamento eficaz geralmente não são conhecidas com antecedência. Assim, aborda-se a variante básica do PRE sob a consideração de que o suprimento de produto do fornecedor e as demandas dos clientes são incertas. Uma formulação de programação estocástica de dois estágios bem como um método de solução heurístico para esse problema são apresentados. Baseados nos resultados dos experimentos computacionais desenvolvidos, mostra-se os mecanismos de resposta das soluções ótimas sob diferentes níveis de incerteza e configurações de custo. Os resultados também mostram que o método heurístico é capaz de resolver instâncias com um grande número de cenários.

Dadas as diferentes variantes práticas estudadas e os métodos de solução especificamente desenvolvidos para essas variantes, essa tese aborda problemas que surgem em vários contextos logísticos práticos e mostra a adaptabilidade da variante básica do PRE e como ela pode ser usada como base para estudar PREs mais ricos. Ela traz contribuições para a literatura científica de otimização da cadeia de suprimentos, assim como para o desenvolvimento de ferramentas para apoiar a tomada de decisões na prática.

Palavras-chave: roteamento de estoques; *branch-and-cut*; metaheurísticas; métodos híbridos; perecibilidade do produto; programação estocástica.

Resumen

La gestión de inventarios y la planeación de la distribución son actividades esenciales para un desempeño eficiente en la cadena de suministro, especialmente para empresas que operan bajo el modelo de inventario administrado por el proveedor. Bajo este modelo de negocios, los proveedores pueden adminisitrar los niveles de inventario y la emisión de órdenes de compra de sus propios clientes, con el objetivo de reducir costos logísticos y mejorar el desempeño de la cadena de suministro. En este contexto, cuando la gestión de inventarios y la planeación de la distribución son tratadas de forma integrada aparece un problema de optimización desafiante, conocido como el problema de ruteo de inventarios (PRI). En el PRI, un proveedor debe determinar de forma simultánea el plan de reabastecimiento para sus clientes en un horizonte de planeación definido y la programación de las rutas de entrega en cada periodo, de modo que una medida de desempeño determinada sea optimizada.

La optimización integrada de las atividades de la gestión de inventarios y planeación de la distribución puede brindar ventajas competitivas para las empresas. Sin embargo, a pesar de su atractivo práctico y de los beneficios sustanciales que esa optimización puede proveer, el PRI recibió una atención creciente sólo en los últimos años. Así, aún existen algunas brechas en relación a la investigación de métodos de solución específicos para variantes prácticas y relevantes de ese problema. Por lo tanto, el objetivo de esta tesis es desarrollar modelos matemáticos exhaustivos y métodos de solución eficaces para diversos PRIs, considerando variantes prácticas para hacer los problemas abordados lo más realista posible.

En primer lugar, la variante básica del PRI es descrita y es presentada una formulación matemática para esa variante. Dos algoritmos metaheurísticos, basados en búsqueda local iterada y recocido simulado, son presentados para resolver la variante básica del PRI considerando dos funciones objetivo diferentes. Los resultados de experimentos computacionales usando instancias de la literatura muestran que los dos algoritmos metaheurísticos pueden encontrar soluciones de alta calidad en tiempos relativamente cortos para ambas funciones objetivo. Adicionalmente, las metaheurísticas propuestas encontraron nuevas mejores soluciones para algunas de las instancias utilizadas.

Luego, una variante práctica del PRI considerando un producto perecedero es estudiada. El estudio del perecimiento de los productos tiene una relevancia significativa en el contexto de la cadena de suministro dado que, en muchas industrias, las materias primas así como los produtos intermedios y finales, pueden ser perecederos. Además, el perecimiento de productos puede aparecer en más de un eslabón de la cadena de suministro. En la variante estudiada, se asume que el producto tiene una vida útil predefinida así como un precio de venta y costo de inventario que dependen de su edad. Para esta variante, son presentadas cuatro formulaciones matemáticas y algoritmos del tipo *branch-and-cut* para resolverlas. Además, una heurística híbrida basada en la combinación de una metaheurística de búsqueda local iterada y dos componentes de programación matemática es presentada. Los resultados de los experimentos computacionales muestran las diferentes ventajas de las formulaciones y la capacidad del método híbrido para lidiar con esta variante, así como con la variante básica del problema.

Finalmente, una variante estocástica del PRI es abordada. Las incertidumbres desempeñan un papel crucial en la gestión de la cadena de suministro dado que informaciones críticas, necesarias para una planeación eficaz de sus actividades, pueden no ser conocidas con antecedencia. Así, se estudia la variante básica del PRI bajo la consideración de que el suministro del producto del proveedor y las demandas de los clientes son inciertas. Para este problema, una formulación de programación estocástica de dos etapas y un método de solución heurístico para ese problema son presentados. Con base en los resultados de experimentos computacionales, se muestran los mecanismos de respuesta de las soluciones óptimas bajo diferentes niveles de incertidumbre y configuraciones de costo. También, los resultados muestran que el método heurístico es capaz de resolver instancias considerando un gran número de escenarios.

Dadas las diferentes variantes prácticas estudiadas y los métodos de solución específicamente desarrollados para cada variante, esta tesis aborda problemas que surgen en varios contextos logísticos prácticos y muestra la adaptabilidad de la variante básica del PRI y como ella puede ser usada como base para estudiar PRIs más ricos. Esta tesis contribuye para el enriquecimiento de la literatura en el área de optimización de la cadena de suministro así como para el desarrollo de herramientas para apoyar en la toma de decisiones en la práctica.

Palabras clave: ruteo de inventarios; *branch-and-cut*; metaheurísticas; métodos híbridos; producto perecedero; programación estocástica.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Context

Inventory management and distribution planning are two activities that are strongly related in the supply chain management (see Figure 1.1). In this context, an effective distribution plan should take into account product availability at the different facilities of the distribution network. On the other hand, inventory management decisions have to adequately synchronize with the distribution plan in order to ensure feasibility and efficiency in the latter operation. The relation between these activities is particularly important in the context of the vendor-managed inventory business model, in which suppliers are allowed to manage the inventory levels and purchasing orders of their customers with the aim of reducing logistics costs and improving the supply chain performance. In addition, suppliers must ensure a minimum service level (e.g., no stockouts) to their customers. The inventory routing problem (IRP) models this situation in an integrated manner by combining inventory management and distribution planning decisions into a single problem (see Figure 1.2). In the IRP, a supplier hast to simultaneously determine the replenishment plan for its customers as well as the periodic routing schedule, such that a given objective (e.g., the sum of routing and inventory holding costs) is optimized while satisfying the demand of its customers during a given planning horizon. This problem can be used as a basis for modeling several practical applications, ranging from offshore transportation in the crude oil and gas industries (Christiansen et al., 2007) to road-based distribution in the bulk gas industry (Singh et al., 2015).

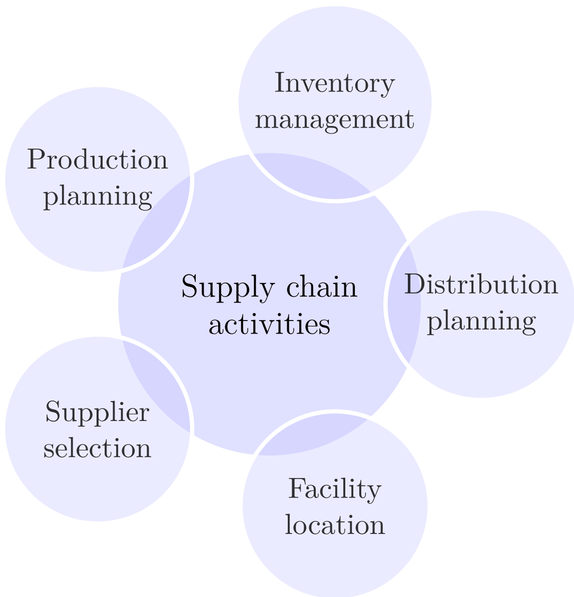


Figure 1.1: Main supply chain planning operations

Managing several supply chain activities through a coordinated approach can help companies to compete more efficiently, streamline their operations, and identify key links in the chain. In addition, the integrated operation of these activities can provide long-term insights for the management of the supply chain (Slack et al., 2010). Furthermore, given the increasing competitiveness of today's business environment, companies have to use their resources as efficiently as

Figure 1.2: Inventory routing problem activities

possible and improve their service levels, while reducing order fulfillment times and inventory levels (Díaz-Madroñero et al., 2015). However, the integrated planning of these operations often poses a major challenge for decision makers given that complex optimization problems arise, such as the IRP. Solving these problems requires state-of-the-art tools and comprehensive analyses, therefore it is essential that decision-making in this context be supported by quantitative approaches, particularly those provided by operations research.

Given the practical appeal of the IRP and the significant benefits that it can bring in real-life applications, as well as its challenging nature, the IRP has attracted the attention of practitioners and researchers over the past decades. Since the pioneering paper of Bell et al. (1983), who solved an integrated inventory management and vehicle routing problem for the distribution of industrial gases, several IRP variants and solution methods for these problems have appeared in the literature. However, since the attention to this type of problems has increased mostly in recent years, there is still room for the development of specific solution methods to address IRP variants with practical relevance. Schmid et al. (2013) and Díaz-Madroñero et al. (2015) pointed out that there exists a considerable lack of research regarding mathematical models and solution approaches for problems integrating activities such as inventory management, production planning, vehicle routing, among others. Therefore, in this thesis we investigate the IRP focusing on the development of mathematical models and solution methods for the basic variant of the IRP as well as relevant practical variants of the problem.

The research developed in this thesis is structured within the context of operations research, with active use of mathematical modeling and computer programming techniques. According to Morabito and Pureza (2012) and Bertrand and Fransoo (2002), the development of this thesis can be characterized in the framework of normative axiomatic quantitative research, since it aims at developing methods and strategies to improve upon the results for a problem previously stated in the literature.

1.2 Organization and contributions

In this section we present the organization of this thesis, briefly describing the contents of each chapter and highlighting their main contributions.

In Chapter 2, we describe the basic variant of the IRP and present a mathematical formulation for this problem. The formulation is based on decision variables that describe the flow of the vehicles on each arc of the distribution network used to represent the problem. We present this formulation using an exponentially large number of subtour elimination constraints, as well as in a compact form. We also discuss the different advantages and disadvantages of these forms for the formulation. We describe this variant of the IRP in a separate chapter since it is used as a basis for the variants explored in the subsequent chapters.

Chapter 3 presents two metaheuristic algorithms based on iterated local search and simulated annealing to solve the basic variant of the IRP. We address this variant under two different objective functions. The first is the standard minimization of the total transportation and inventory holding costs, while the second is the ratio between total transportation costs and total quantity delivered to the customers, called logistic ratio. This latter objective function can be more realistic in some logistics settings, even though it represents an additional challenge to exact methods given its nonlinear nature. The results of extensive computational experiments using instances from the literature show that the two presented metaheuristic algorithms can effectively handle both objective functions, providing high-quality solutions within relatively short running times. In addition, the metaheuristics were able to find new best solutions for some of the benchmark instances.

In Chapter 4 we study an IRP in which goods are perishable. In this problem, the product is assumed to have a fixed shelf-life during which it is usable and after which it must be discarded. Age-dependent revenues and inventory holding costs are also considered. We introduce four mathematical formulations for the problem, two with a vehicle index and two without it, and present branch-and-cut algorithms to solve them. In addition, we propose a hybrid heuristic based on the combination of an iterated local search metaheuristic and two mathematical programming components. We analyze the results of extensive computational experiments using problem instances from the literature as well as new larger instances. The results indicate the different advantages of the introduced formulations and show that the hybrid method is able to provide high-quality solutions in relatively short running times for small- and medium-sized instances, while good quality solutions are found within reasonable running times for larger instances. We also adapt the proposed hybrid heuristic to solve the basic variant of the IRP. The results using standard instances show that our heuristic is also able to find good quality solutions for this problem when compared to the state-of-the-art methods from the literature.

Chapter 5 focuses on a stochastic variant of the IRP under the consideration that both the product supply and the customer demands are uncertain. We propose a two-stage stochastic programming formulation where routing decisions are made in the first stage, while delivery quantities, inventory levels and specific recourse actions are made in the second stage. This formulation can be adapted to consider different recourse mechanisms, such as lost sales, backlogging and an additional supply source in a capacity reservation contract setting. The objective is to minimize the first-stage cost plus the total expected inventory and recourse cost incurred in the second stage. We also present a heuristic solution method based on the progressive hedging algorithm. We provide managerial insights resulting from extensive computational experiments using instances generated from a benchmark test set of the literature. In particular, we study the response mechanisms of the optimal solutions under different uncertainty levels of the random variables and different cost configurations. The results with the heuristic method show that it provides high-quality solutions within reasonable running times for instances with a large number of scenarios.

Finally, in Chapter 6 we present an overall final discussion and concluding remarks together with perspectives for future research arising from the developments presented in this thesis.

Chapter 2

The inventory routing problem

In this chapter, we describe the basic variant of the IRP and present a mathematical formulation for the problem. The formulation uses variables describing the flow of the vehicles on the arcs of the graph used to represent the problem. We present this formulation in two different forms, namely: (*i*) using an exponentially large number of subtour elimination constraints, and (*ii*) in a compact form using MTZ-like subtour elimination constraints. This variant of the IRP is described in a separate chapter given that it is used as a basis for all the variants addressed in this thesis.

2.1 The basic variant of the inventory routing problem

In the basic variant of the IRP, a single supplier is responsible for delivering a single product to a set of customers over a finite planning horizon. Under the assumption that the travel costs are symmetric, the problem can be defined on a complete undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \{0, 1, \ldots, N\}$ is the vertex set and $\mathcal{E} = \{(i, j) \colon i, j \in \mathcal{N}, i < j\}$ is the edge set. Vertex 0 represents the supplier depot which has a homogeneous fleet of K vehicles of capacity Q each, denoted by set $\mathcal{K} = \{1, \ldots, K\}$. The remaining vertices of set $\mathcal{N}$, denoted by $\mathcal{C} = \{1, \ldots, N\}$, represent the customers. Therefore, the vertex set $\mathcal{N}$ represents all the facilities of the distribution network.

The planning horizon is denoted by a set of time periods $\mathcal{T} = \{1, \ldots, T\}$. A travel cost c_{ij} is associated with every edge $(i, j) \in \mathcal{E}$ and inventory holding costs h_i^t are charged at both the supplier 0 and the customers $i \in \mathcal{C}$ for each unit of product at the end of every time period. Each customer $i \in \mathcal{C}$ has a limited storage capacity C_i and each facility $i \in \mathcal{N}$ has an initial inventory I_i^0. Each customer $i \in \mathcal{C}$ has a known demand d_i^t for the product in every time period $t \in \mathcal{T}$, which is the minimum amount of product that the supplier must guarantee to be available at the customer at that time period. In addition, the supplier produces or receives a quantity r^t of the product in each time period $t \in \mathcal{T}$. Table 2.1 summarizes all the notation previously introduced.

Sets:	
$\mathcal{C}$	Set of customers
$\mathcal{N}$	Set of vertices/facilities
$\mathcal{E}$	Set of edges
$\mathcal{T}$	Set of time periods
$\mathcal{K}$	Set of vehicles
Parameters:	
h_i^t	Inventory holding cost at facility i at the end of time period t
c_{ij}	Transportation cost between facilities i and j
d_i^t	Demand of customer i in time period t
r^t	Amount made available at the supplier in time period t
C_i	Storage capacity of customer i
I_i^0	Initial inventory at facility i
Q	Capacity of each vehicle

Table 2.1: Sets and parameters of the problem

The basic variant of the IRP consists of determining the time periods in which the customers will be visited; the quantity of product that will be delivered in every visit; and the delivery routes to perform those visits. The objective is to minimize the total cost, given by the sum of inventory holding and routing costs. The holding costs are charged on the inventories at the end of every time period at both the supplier and customers. It is assumed that the supplier holding capacity is unbounded. In addition, according to the usual practice in the literature, it is also assumed that the customers who receive a delivery in a given time period can use this to fulfill the demand in the same time period. Also, the amount made available in each time period at the supplier can be used for deliveries in the same time period. Figure 2.2 shows a graphical representation of the order of these events.

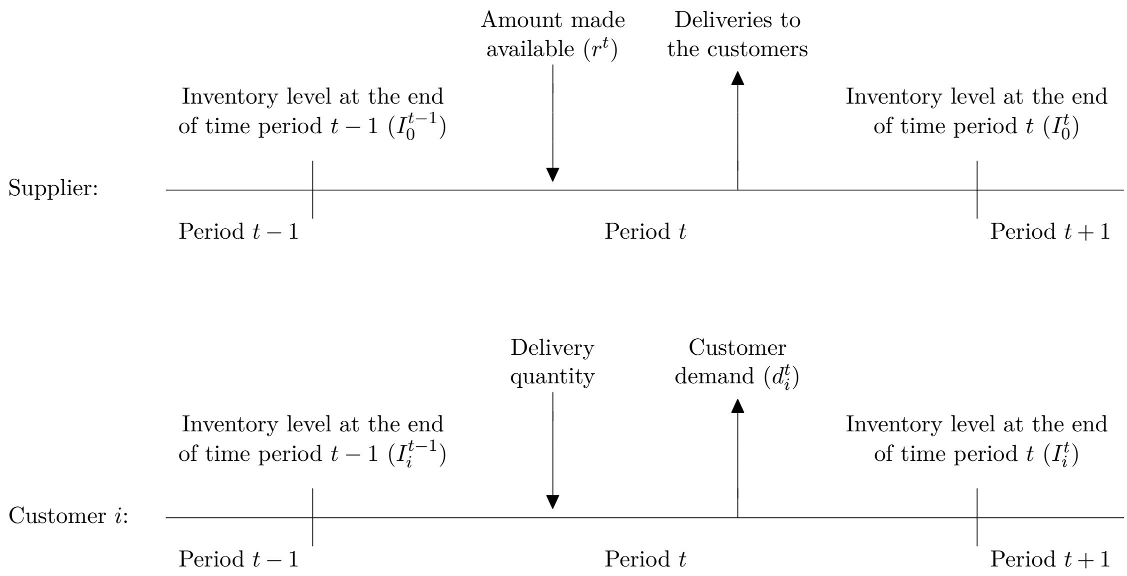


Table 2.2: Timing of the operations in the IRP

The IRP can be used to model many different applications in practice, such as bulk gas distribution (Singh et al., 2015); oil and gas distribution in maritime applications (Song and Furman, 2013); fuel delivery to filling stations (Popović et al., 2012); vending machines replenishment (Huang and Lin, 2010); and ATM cash replenishment (Larrain et al., 2017). Given these multiple applications, no standard variant of the problem has be defined. Then, since the variant just described corresponds to a simplified case that does not include any specific practical constraint neither any restricting assumption, we call it the 'basic variant' of the IRP (Coelho et al., 2014b).

2.2 An arc-based formulation for the inventory routing problem

To model this variant of the IRP using arc variables, as the ones presented by Archetti et al. (2007) and Coelho and Laporte (2013b), consider the following notation. Let $U_i^t = \min\{Q, C_i\}$ be an upper bound on the amount that can be delivered to customer i in time period t. Finally, consider the following decision variables:

$$x_{ij}^{kt} \in \{0,1,2\} : \text{number of times vehicle } k \in \mathcal{K} \text{ traverses edge } (i,j) \in \mathcal{E} \text{ in time period } t \in \mathcal{T};$$
$$y_i^{kt} \in \{0,1\} : 1 \text{ if facility } i \in \mathcal{N} \text{ is visited by vehicle } k \in \mathcal{K} \text{ in period } t \in \mathcal{T}, 0 \text{ otherwise;}$$
$$I_i^t \geq 0 : \text{inventory level at facility } i \in \mathcal{N} \text{ at the end of time period } t \in \mathcal{T};$$
$$q_i^{kt} \geq 0 : \text{quantity delivered to customer } i \in \mathcal{C} \text{ by vehicle } k \in \mathcal{K} \text{ in time period } t \in \mathcal{T}.$$

Notice that the routes start from the supplier facility and thus the variable y_0^{kt} represents whether or not the vehicle $k \in \mathcal{K}$ is used in time period $t \in \mathcal{T}$. Given these variables, we first present the arc-based formulation using an exponentially large number of subtour elimination constraints (SECs), as follows:

$$
\text{(OF1)} \quad \min \sum_{(i,j)\in\mathcal{E}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} c_{ij} x_{ij}^{kt} + \sum_{i\in\mathcal{N}} \sum_{t\in\mathcal{T}} h_i^t I_i^t \tag{2.1}
$$

$$
\text{s.t.} \quad I_0^t = I_0^{t-1} + r^t - \sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}} q_i^{kt} \qquad t\in\mathcal{T}, \tag{2.2}
$$

$$
I_i^t = I_i^{t-1} + \sum_{k\in\mathcal{K}} q_i^{kt} - d_i^t \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{2.3}
$$

$$
I_i^{t-1} + \sum_{k\in\mathcal{K}} q_i^{kt} \leq C_i \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{2.4}
$$

$$
q_i^{kt} \leq U_i^t y_i^{kt} \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{2.5}
$$

$$
\sum_{i\in\mathcal{C}} q_i^{kt} \leq Q y_0^{kt} \qquad k\in\mathcal{K}, t\in\mathcal{T}, \tag{2.6}
$$

$$
\sum_{j\in\mathcal{N}:j<i} x_{ji}^{kt} + \sum_{j\in\mathcal{N}:i<j} x_{ij}^{kt} = 2y_i^{kt} \qquad i\in\mathcal{N}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{2.7}
$$

$$
\sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{S}:j>i} x_{ij}^{kt} \leq \sum_{i\in\mathcal{S}} y_i^{kt} - y_\ell^{kt} \qquad \forall\mathcal{S}\subseteq\mathcal{C}, |\mathcal{S}|>2, k\in\mathcal{K}, t\in\mathcal{T}, \ell\in\mathcal{S}, \tag{2.8}
$$

$$
\sum_{k\in\mathcal{K}} y_i^{kt} \leq 1 \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{2.9}
$$

$$
I_i^t \geq 0 \qquad i\in\mathcal{N}, t\in\mathcal{T}, \tag{2.10}
$$

$$
q_i^{kt} \geq 0 \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{2.11}
$$

$$
y_i^{kt} \in \{0,1\} \qquad i\in\mathcal{N}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{2.12}
$$

$$
x_{ij}^{kt} \in \{0,1\} \qquad (i,j)\in\mathcal{E}: i\neq 0, k\in\mathcal{K}, t\in\mathcal{T}, \tag{2.13}
$$

$$
x_{ij}^{kt} \in \{0,1,2\} \qquad (i,j)\in\mathcal{E}: i=0, k\in\mathcal{K}, t\in\mathcal{T}. \tag{2.14}
$$

The objective function (2.1), which we will refer to as OF1, consists of minimizing the total cost, given by the sum of transportation and inventory holding costs. Constraints (2.2) and (2.3) define the inventory balance at the supplier and at the customers, respectively. Constraints (2.4) impose that the inventory level after delivery at the customer facilities cannot exceed their respective storage capacity. Constraints (2.5) permit a vehicle to perform a delivery to a specific customer only if this customer is visited by the vehicle. Constraints (2.6) guarantee that the capacity of each vehicle is respected. Constraints (2.7) ensure the vehicle flow conservation. Constraints (2.8) are SECs, defined for each possible subset of customers. Constraints (2.9) define that each customer can be visited at most once in each time period. Finally, the domain of the decision variables is defined in constraints (2.10)-(2.14). Notice that when $i \neq 0$ and $j > i$, x_{ij}^{kt} can only take the values 0 or 1; if $i = 0$, then x_{ij}^{kt} can also be equal to 2, indicating that vehicle k makes a round trip between the depot and customer j in time period t.

This type of formulation, containing an exponentially large number of SECs, typically provides stronger bounds than their respective compact counterparts (Öncan et al., 2009). However, they usually require specialized separation procedures implemented within branch-and-cut schemes, as we will show in Chapters 4 and 5. On the other hand, compact formulations have the advantage of being easily implementable using general-purpose optimization softwares such as

CPLEX (IBM ILOG CPLEX, 2009) or Gurobi (Gurobi Optimization, Inc., 2015). Therefore, we also present a compact version of the arc-based formulation using MTZ-like SECs (Miller et al., 1960). To formulate the problem using a compact formulation, consider the following additional notation. Let $\mathcal{A} = \{(i,j)\colon i,j \in \mathcal{N}, i \neq j\}$ be the arc set. Consider also the continuous variable u_i^{kt} that accumulates the number of customers visited by the route of vehicle $k \in \mathcal{K}$ in period $t \in \mathcal{T}$ after the visit to facility $i \in \mathcal{N}$. This variable is used in the SECs of the formulation. Also, now x_{ij}^{kt} is a binary variable indicating whether or not arc $(i,j) \in \mathcal{A}$ is traversed by vehicle $k \in \mathcal{K}$ in time period $t \in \mathcal{T}$. It is worth mentioning that in this formulation we use the arc set $\mathcal{A}$ instead of the edge set $\mathcal{E}$ given the cumulative nature of variables u. Given this notation, the compact version of the arc-based formulation is presented below:

$$\min \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} c_{ij} x_{ij}^{kt} + \sum_{i\in\mathcal{N}} \sum_{t\in\mathcal{T}} h_i^t I_i^t \tag{2.15}$$

$$\text{s.t.} \quad (2.2)\text{-}(2.6), (2.9)\text{-}(2.12)$$

$$\sum_{j\in\mathcal{N}:j\neq i} x_{ij}^{kt} = y_i^{kt} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{2.16}$$

$$\sum_{j\in\mathcal{N}:j\neq i} x_{ji}^{kt} = y_i^{kt} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{2.17}$$

$$u_j^{kt} \geq u_i^{kt} + 1 - N(1 - x_{ij}^{kt}) \qquad (i,j) \in \mathcal{A}\colon j \neq 0, k \in \mathcal{K}, t \in \mathcal{T}, \tag{2.18}$$

$$u_i^{kt} \geq 0 \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{2.19}$$

$$x_{ij}^{kt} \in \{0,1\} \qquad (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{2.20}$$

The objective function (2.15) (equivalent to OF1) consists of minimizing the sum of transportation and inventory holding costs. Constraints (2.16) and (2.17) ensure the vehicle flow conservation. Constraints (2.18) define that variables u accumulate at least the number of facilities visited for each vehicle in each time period. These constraints eliminate subtours not containing the supplier. Finally, the domain of the decision variables is defined in constraints (2.19)-(2.20).

Notice that under the assumption of identical vehicles in terms of capacity as well as travel times and costs, the IRP can also be modeled using a formulation without a vehicle index. For formulation (2.1)-(2.14), capacity cuts in the form shown in (2.21) would need to be included as SECs and capacity constraints.

$$Q \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{S}:i<j} x_{ij}^t \leq Q \sum_{i\in\mathcal{S}} y_i^t - \sum_{i\in\mathcal{S}} q_i^t \quad \forall \mathcal{S} \subseteq \mathcal{C}, |\mathcal{S}| \geq 2, t \in \mathcal{T}. \tag{2.21}$$

This formulation has the advantage of having fewer integer variables when compared to the formulation with a vehicle index. However, it cannot be used for cases with heterogeneous fleet, which makes it less extensible. Also notice that all the mathematical formulations presented in this section can be reformulated without using visit decision variables (y). Nevertheless, this could negatively affect the performance of the algorithms used to solve them because these variables are useful in the branching process. Branching on the visit variables y (instead of on the vehicle flow variables x) implies a focus on determining in which periods each customer must be visited (instead of which arcs must be used). Such a branching strategy can be very effective as this can significantly improve the lower bounds in many nodes of the branch-and-cut tree (Desaulniers et al., 2016).

In addition, it is worth mentioning that the IRP can also be represented using column generation-based formulations. Although this type of formulation requires more sophisticated solution techniques than their network flow-based counterparts, they usually provide considerably tighter bounds, which yields promising results when used for IRPs (Engineer et al., 2012; Desaulniers et al., 2016), similar to those observed for vehicle routing problems (Desaulniers et al., 2008; Munari and Gondzio, 2013; Pecin et al., 2014; Alvarez and Munari, 2017). Thus, this type of formulation can be used to devise efficient branch-and-price or column generation-based solution methods.

Notice also that alternatively to the objective function defined in (2.1) (also in (2.15)), different objective functions can be used in the IRP context. For example, omitting the inventory holding costs, as in some cases in the maritime industry (only transportation cost and the loading/unloading costs, which include port operations, duties, etc., are considered) (Al-Khayyal and Hwang, 2007; Engineer et al., 2012; Stanzani et al., 2018) or in the bulk gas industry (only transportation cost) (Campbell and Savelsbergh, 2004; Savelsbergh and Song, 2008). In addition, an alternative objective function has been addressed recently in the literature, based on the logistic ratio (Benoist et al., 2011; Singh et al., 2015; Archetti et al., 2017b, 2019). This criterion consists of minimizing the average cost to distribute one unit of product (OF2), as follows:

$$\text{(OF2)} \quad \min \frac{\sum_{(i,j)\in\mathcal{E}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} c_{ij} x_{ij}^{kt}}{\sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} q_i^{kt}}. \tag{2.22}$$

While more realistic in some logistics settings, this objective function represents an additional challenge to exact methods given that it is a nonlinear function.

Finally, it is worth highlighting that in the formulations previously presented, the quantities delivered follow the maximum-level replenishment policy. In this policy, the delivery quantities are flexible and bounded only by the actual capacity of the customer and the capacity of the vehicle. However, different replenishment policies can be used, e.g., the order-up-to level which defines that whenever a customer is visited, the quantity delivered is such that the customer inventory level reaches its maximum level. In addition, the models can reformulated using delivery variables (q) indicating the detailed use of the deliveries, as in the facility location formulation of the single item uncapacitated lot sizing problem (Krarup and Bilde, 1977), as will be shown in Chapter 4.

Chapter 3

Metaheuristic algorithms for the basic variant of the IRP

This chapter presents two metaheuristic algorithms based on iterated local search and simulated annealing to solve the basic variant of the IRP. We address this variant under two different objective functions. The first one is the standard minimization of the total transportation and inventory holding costs while the second is the ratio between total transportation costs and the total quantity delivered to the customers, called logistic ratio. Computational experiments show that these algorithms provide reasonably high quality solutions in relatively short running times for both objective functions when compared to other methods for well-known instances from the literature. Moreover, the algorithms produce new best solutions for some of these instances. It is worth mentioning that the mathematical notation used in this chapter is the same defined in Chapter 2.

⋆ An article based on the contents of this chapter is published as:

Alvarez, A., Munari, P., and Morabito, R. (2018). Iterated local search and simulated annealing algorithms for the inventory routing problem. *International Transactions in Operational Research*, 25:1785-1809.

3.1 Introduction

Since the paper of Bell et al. (1983) solving an integrated inventory management and vehicle routing problem for the distribution of industrial gases, different IRPs have been studied in the literature. For instance, Dror et al. (1985) addressed a long-term IRP using a rolling horizon strategy; later, Dror and Ball (1987) solved the same problem by reducing the planning horizon to a single-period problem, defining costs reflecting long-term decisions, safety inventory levels and subsets of customers to be considered. Campbell and Savelsbergh (2004) developed a two-phase hybrid approach for an IRP minimizing transportation costs only. In their approach, the visit schedule and delivery sizes are determined solving an integer linear model while the delivery routes are constructed with heuristic algorithms. Savelsbergh and Song (2007, 2008) focused on the IRP with continuous moves, motivated by a real-life application that includes pickup and delivery routes spanning multiple time periods. The authors developed heuristic and hybrid algorithms to solve the problem. Archetti and Speranza (2016) showed the importance of the IRP by comparing the heuristic solution of the IRP with the solution obtained by sequentially solving to optimality an inventory management problem and then a vehicle routing problem.

Other extensions of the IRP have been addressed by Le et al. (2013), who used a column-generation based heuristic to solve an IRP with perishability constraints. Cordeau et al. (2015) solved the multi-product IRP with a three-phase decomposition-based heuristic algorithm. Abdelmaguid et al. (2009) proposed heuristic algorithms for an IRP with backlogging. Shiguemoto and Armentano (2010) developed a Tabu Search metaheuristic for an integrated production-distribution problem, which was also applied to the IRP. Benoist et al. (2011) and Singh et al. (2015) solved different IRPs by applying heuristic algorithms to real-life distribution problems in the bulk gas industry. For other studies on applications and variants of the IRP, see the comprehensive reviews by Andersson et al. (2010) and Coelho et al. (2014b).

Regarding solution approaches for the basic variant of the IRP (2.1)-(2.14), hybrid methods combining metaheuristic algorithms with mathematical programming have been proposed by Archetti et al. (2012), Coelho et al. (2012), Adulyasak et al. (2014b) and Santos et al. (2016). Exact methods have also been presented in recent years, based on branch-and-cut (B&C) and branch-price-and-cut (BPC) algorithms (Archetti et al., 2007; Solyalıand Süral, 2011; Coelho and Laporte, 2013a, 2014a; Archetti et al., 2014; Desaulniers et al., 2016). As observed in the computational results presented in these papers, exact approaches are able to solve only small-to medium-sized instances within running times acceptable in practice.

Recently, Archetti et al. (2017b) presented a BPC-based method to address the IRP with the logistic ratio as objective function, as Equation (2.22). The logistic ratio can be of great value in practical contexts, as it can better reflect the efficiency of the distribution process. Despite its practical motivation, only a few studies have addressed this kind of objective function in the context of IRPs (Benoist et al., 2011; Singh et al., 2015; Archetti et al., 2017b).

In this chapter, we develop two metaheuristic algorithms based on iterated local search (ILS) and simulated annealing (SA) to solve the basic variant of the IRP. ILS combines a local search heuristic with a perturbation algorithm to escape from local optimal solutions, whereas SA can probabilistically accept solutions that temporarily produce degradations in the current search point to avoid getting trapped in local optimal solutions. The metaheuristics are also adapted to solve the IRP with the logistic ratio as alternative objective function.

The main contributions of this chapter are twofold. First, we present two metaheuristic algorithms capable of generating high-quality solutions in relatively short running times for the IRP if compared to other solution methods of the literature. In particular, new best solutions were found for some of the benchmark instances. Second, we address an alternative objective function that minimizes the logistic ratio, which is more realistic in some logistics settings. We are not aware of any other heuristic method that addresses this objective in this IRP variant. The goal is to verify if the proposed methods are also effective with this alternative objective function in comparison with the best available results. The overall results show that ILS and SA can also have a good performance when solving the given IRP variant.

This chapter is organized as follows. In Section 3.2, we describe a construction heuristic for the basic variant of the IRP. Sections 3.3 and 3.4 present in detail the components of the ILS and SA metaheuristics algorithms developed to solve the problem, respectively. Finally, in Section 3.6 we describe the results of the computational experiments performed with the metaheuristic algorithms.

3.2 A construction heuristic for the basic variant of the IRP

We devise a construction heuristic that separates the decisions of the problem into two phases in a iterative scheme. In the first phase of each iteration, the heuristic defines a set of customers for which a visit is mandatory or potentially profitable. For these customers, the amount of product that will be delivered is also defined in this stage. Then, the second phase consists of the definition of feasible delivery routes for visiting the customers selected in the first phase.

The heuristic is strongly based on the inventory levels I_i^t of each customer $i \in \mathcal{C}$ at each time period $t \in \mathcal{T}$. At the beginning of any iteration of the heuristic, the inventory levels can be computed using the initial inventory level (I_i^0), the demands (d_i^t), and all the deliveries received by the customer in previous periods (q_i^t), as follows:

$$I_i^t = I_i^0 - \sum_{h=1}^{t} d_i^h + \sum_{h=1}^{t-1} q_i^h, \quad \forall i \in \mathcal{C}, t \in \mathcal{T}. \tag{3.1}$$

Inventory levels are first computed at the beginning of the heuristic, which starts from the first time period of the planning horizon. Since there are no product deliveries up to this point, they are initialized as

$$I_i^t = I_i^0 - \sum_{h=1}^{t} d_i^h, \quad \forall i \in \mathcal{C}, t \in \mathcal{T}. \tag{3.2}$$

Then, these levels are updated at the end of each iteration based on the deliveries of the routes obtained in the iteration.

There is one iteration of the heuristic for each time period $t \in \mathcal{T}$, starting from $t = 1$. In the first phase of iteration t, the heuristic selects some customers and separates them into two sets,

$\mathcal{C}_1$ and $\mathcal{C}_2$. The first set ($\mathcal{C}_1$) is composed of those customers for which a stockout will occur if they are not visited in the current time period, i.e., those with $I_i^t < 0$. Since no stockout is allowed in a solution, these customers are included in the first set and then the heuristic decides the delivery quantity as follows. If it is the last time period (i.e., $t = T$), then the delivery quantity is defined according to Equation (3.3), which states that the delivery quantity is fixed to the minimum possible amount that can be delivered to the customer to avoid stockout in the current time period, respecting the capacity of the vehicle.

$$q_i^t = \min\{-I_i^t, Q\}, \tag{3.3}$$

On the other hand, if the current iteration does not correspond to the last time period, i.e., $t < T$, the delivery quantity is set to the effective capacity at that time point, given by difference between the customer capacity and the inventory level on the previous time period. In this case, the vehicle capacity is also considered, as stated in Equation (3.4).

$$q_i^t = \min\{C_i - I_i^{t-1}, Q\}. \tag{3.4}$$

The remaining customers (i.e., those with $I_i^t \geq 0$) are selected to enter in the second set ($\mathcal{C}_2$) according to the following rules. First, their respective delivery quantities are defined similarly to (3.4), but the first term ($C_i - I_i^{t-1}$) is multiplied by the parameter `ratio_demand` $\in (0, 1]$, which defines the proportion of the maximum possible quantity that will be actually delivered. Then, customer i is inserted into the second set based on the following criteria:

1. If $t < T$, then we analyze the urgency degree of the customer. In this stage we insert the customer into the second set if a stockout may occur in the next `look_ahead` time periods, that is, $I_i^{t'} < 0$ for any $t' = t+1, \ldots, t+$`look_ahead`. The value of `look_ahead` determines how far to look forward in the planning horizon to anticipate forthcoming stockouts and perform an early delivery;

2. On the other hand, if $t = T$, then we analyze the potential profitability of the visit. The customer will be included in the second set if the cost of keeping the product at the supplier in period t is greater than the cost of keeping it at the customer plus an estimate of the delivery cost given by the cost of a round trip to the customer from the depot, that is, the customer is inserted if $q_i^t h_0^t > q_i^t h_i^t + 2c_{0i}$.

After this first phase, the second phase of the iteration starts. It consists of determining one or more vehicle routes, based on the delivery quantities defined in the previous phase. We used a standard nearest-neighbor insertion heuristic (Bräysy and Gendreau, 2005), that first routes customers from set $\mathcal{C}_1$ (mandatory customers) and then tries to insert the customers from set $\mathcal{C}_2$ into the route as long as the insertion satisfies the vehicle capacity. Then, the inventory levels of the customers routed in this phase are updated given the performed deliveries q_i^t in the current time period. If the updated level I_i^t of at least one customer in the first set remains negative, then the heuristic terminates with no solution. Otherwise, a new iteration is started for the next time period ($t + 1$), until the end of the planning horizon.

A pseudo-code of this heuristic is shown in Algorithm 3.1. As observed in preliminary experiments, this heuristic can run in very short running times for even large-sized problem instances. Thus, it was embedded within two loops to explore different values for the parameters `ratio_demand` and `look_ahead`, aiming to find the best possible solution. For `ratio_demand` we tried all values in the range $(0, 1]$, starting from 1.0 and reducing 0.1 at each iteration. For `look_ahead`, values between one and half of the size of the planning horizon ($\lceil T/2 \rceil$) are tested. This heuristic was used as source of initial solutions in both metaheuristic algorithms. As will be shown in Section 3.6, this heuristic was capable of finding feasible solutions for all the instances used in this chapter (almost 1100 problem instances).

Algorithm 3.1: Construction heuristic

begin
  $S^* \leftarrow \emptyset$;
  `ratio_demand` $\leftarrow 1.0$;
  while `ratio_demand` > 0 **do**
    `look_ahead` $\leftarrow 1$;
    while `look_ahead` $\leq \lceil T/2 \rceil$ **do**
      Compute the inventory levels I_i^t as in (3.2), for all $i \in \mathcal{C}$ and $t \in \mathcal{T}$;
      for each time period t in $\mathcal{T}$ **do**
        $\mathcal{C}_1 \leftarrow \emptyset$;
        $\mathcal{C}_2 \leftarrow \emptyset$;
        for each customer i in $\mathcal{C}$ **do**
          if $I_i^t < 0$ (there will be stockout on the period) **then**
            if $t = T$ **then** $q_i^t \leftarrow \min\{-I_i^t, Q\}$;
            else $q_i^t \leftarrow \min\{C_i - I_i^{t-1}, Q\}$;
            Add i to $\mathcal{C}_1$;
          else
            $q_i^t \leftarrow \min\{(C_i - I_i^{t-1}) *$ `ratio_demand`$, Q\}$;
            if criterion 1 or 2 are satisfied **then** Add i to $\mathcal{C}_2$;
          end
        end
        Route customers in $\mathcal{C}_1 \cup \mathcal{C}_2$ using the nearest-neighbor insertion heuristic, such that all customers in $\mathcal{C}_1$ are visited;
        Update I_i^t using the values q_i^t of the deliveries made to each customer i in the obtained route(s), as in (3.1);
      end
      Update best feasible solution S^*;
      `look_ahead` $\leftarrow$ `look_ahead` $+ 1$;
    end
    `ratio_demand` $\leftarrow$ `ratio_demand` $- 0.1$;
  end
end

3.3 Iterated local search metaheuristic

In this section we describe the main features and components of the ILS-based approach proposed in this thesis. ILS is a metaheuristic algorithm that applies two main steps in a iterative scheme. The first step corresponds to a local search phase which aims to reach a local optimal solution and then, in the second step, the solution reached is perturbed by applying random changes to its elements. The latter step is applied with the aim of reaching a new search point from which the algorithm will continue the exploration. These steps are iteratively repeated, leading to a randomized walk in the space of local optimal solutions (Lourenço et al., 2003).

The basic components of an ILS-based algorithm are a construction heuristic to provide the starting point of the exploration, a local search procedure to reach the local optimal solutions, a perturbation mechanism to escape from the local optimal solutions and an acceptance criterion to define whether or not a reached solution should be accepted as the new search point.

A pseudo-code of the proposed approach is shown in Algorithm 3.2. The initial solution is generated using the construction heuristic described in Section 3.2 (line 2). If the construction heuristic cannot find a feasible solution, the algorithm stops; otherwise, the searching process of the metaheuristic starts. A multi-start randomized variable neighborhood descent (RVND) heuristic performs the local search (lines 4 and 7) and a multi-operator algorithm is used as a perturbation mechanism (line 6). These components are described in detail below. Finally, the acceptance criterion keeps the reached solution only if its objective value is better than the current best solution (lines 8-10).

Algorithm 3.2: Iterated local search

```
1  begin
2      S0 ← Construct initial solution;
3      if S0 ≠ ∅ then
4          S* ← localSeach(S0);
5          while stop criterion is not met do
6              S' ← perturb(S*);
7              S' ← localSeach(S');
8              if f(S') < f(S*) then
9                  S* ← S';
10             end
11         end
12     end
13 end
```

The proposed metaheuristic is similar to the ILS-based hybrid method developed by Santos et al. (2016) also for the basic variant of the IRP. Similar local search and perturbation algorithms are used in both of them. However, different to their approach (as will be seen in the following subsections), the ILS proposed in this chapter applies the local search heuristic in a multi-start approach to explore more intensively and effectively the search space. Also, our method does not rely on any mathematical programming component.

3.3.1 Local search heuristic

For the local search procedure, we use a variable neighborhood descent heuristic (Mladenović and Hansen, 1997) with random neighborhood ordering. In a randomized variable neighborhood descent heuristic, local search operators are selected at random and applied to the incumbent solution until none of them can improve it (Subramanian et al., 2010). This non-deterministic behavior of the algorithm helps in the diversification of the metaheuristic, since local optimal solutions can be different for distinct local search operators. Therefore, each time the RVND is applied a different solution can be obtained. In addition, the randomized order leads to a more balanced exploration of the neighborhoods, given that when a fixed sequential order is adopted most of the effort is spent on the first operators (Deng and Bard, 2011). A pseudo-code of the RVND heuristic is shown in Algorithm 3.3. The heuristic starts with an initial solution (line 2) and a set containing the predefined local search operators (line 3). Then, while the set is not empty, an operator is chosen at random (line 5) and applied to the incumbent solution (line 6). If the operator improves the solution, the set is re-established to its initial configuration (line 8). Otherwise, the operator is removed from the set (line 10) and the process continues with the remaining operators. In our implementation, all operators exhaustively explore the search space using the best improvement strategy and only feasible solutions are allowed through the search process.

Algorithm 3.3: Randomized variable neighborhood descent heuristic

```
1  begin
2      S* ← S0 (save initial solution);
3      V ← initialize set of local search operators;
4      while | V |> 0 do
5          v ← select at random a local search operator from V;
6          Apply v to S*;
7          if v improved S* then
8              V ← reinitialize set of local search operators;
9          else
10             remove v from V;
11         end
12     end
13 end
```

In the heuristic, we use seven local search operators. The first three are routing operators as they only change the routes of the solution. The rest are inventory-routing operators since they can change the routes as well as the quantities delivered. They are defined as follows:

- *Or-opt(k)*, $k \in \{1, 2, 3\}$: transfers k adjacent customers from their current position to another in the same route;
- *Shift(k)*, $k \in \{1, 2, 3\}$: relocates k adjacent customers to another route in the same time period;

- *Swap(k_1, k_2)*, $k_1, k_2 \in \{1, 2\}, k_1 \geq k_2$: exchanges k_1 adjacent customers in a route with k_2 adjacent customers in another route in the same time period;

- *Increase/reduce deliveries*: for each customer $i \in \mathcal{C}$, this operator increases/reduces as much as possible the quantities delivered in the visits (q_i^t), based on the profitability of the increase/reduction, given by the difference on the inventory holding costs between the customer and the supplier on the current time period t ($h_0^t - h_i^t$). Thus, for a given customer $i \in \mathcal{C}$ and time period $t \in \mathcal{T}$, if $h_0^t > h_i^t$, the operator tries to increase the delivery performed to the customer in the time period (using $U_i = \{Q, C_i\}$ as upper bound on the delivery quantity); otherwise, the operator attempts to reduce as much as possible the amount delivered;

- *Merge visits*: for each customer $i \in \mathcal{C}$, the operator tries to merge each pair of visits (receiving $q_i^{t_1}$ and $q_i^{t_2}$) that occurs in different time periods ($t_1 \neq t_2$) to a single period (either t_1 or t_2);

- *Transfer visits*: for each customer $i \in \mathcal{C}$, this operator tries to move each visit (receiving q_i^t) to all different time periods $t' \in \mathcal{T}$ in which the customer is not visited ($\forall t' \in \mathcal{T}, t' \neq t$ and $q_i^{t'} = 0$);

- *Insert visits*: for each customer $i \in \mathcal{C}$, this operator tries to add new visits (delivering the maximum possible quantity) in all the time periods $t \in \mathcal{T}$ in which the customer is not visited ($\forall t \in \mathcal{T}, q_i^t = 0$) if the inventory holding cost of the customer is lower than the inventory holding cost of the supplier ($h_0^t > h_i^t$).

The operators, as described above, are used when minimizing OF1 (sum of the costs). For OF2 (logistic ratio) some few changes must be applied, as follows. The *Reduce deliveries* operator is only applied if the delivery can be completely removed, since reducing a delivery maintaining the visit to the customer worsens the value of the logistic ratio. The *Increase deliveries* operator tries to increase all the amounts delivered, not only for those customers with $h_0^t > h_i^t$. Finally, the *Insert deliveries* operator tries to set new visits for all customers, not only for those with $h_0^t > h_i^t$. The last two changes were applied given that the holding costs are not considered in the logistic ratio formula (2.22). Every time an insertion operation is performed in one of the operators, the customer is inserted into the cheapest position of the selected route.

Since RVND is a non-deterministic algorithm (local optimal solutions can be different for distinct local search operators and random ordering of the operators), we used a multi-start RVND heuristic to boost the intensification phase of the metaheuristics and reduce the variability of the results. The starting solution is always the input solution of the operator. The number of times the algorithm repeats the process is controlled by the parameter `maxIterRVND`. The scale of `maxIterRVND` defines the balance between quality of the solutions and computational effort. In addition, we performed preliminary experiments which showed that the randomized version of RVND led, on average, to better results when compared to different versions with deterministic neighborhood ordering. Thus, we decided to maintain the randomized order.

3.3.2 Perturbation mechanism

As previously mentioned, a perturbation mechanism is needed in order to escape from local optimal solutions. The performance of the ILS is strongly related to this mechanism because its strength defines much of the behavior of the metaheuristic. The mechanism must be able to effectively diversify the search process without rendering ILS a random restart search. In our implementation, given the multiple decisions made simultaneously, we propose a perturbation algorithm composed of four different operators, as follows.

- *Random shift*: chooses a route and one of its customers and transfers it to another route in the same period;
- *Random delivery reduction*: selects a route and one of its customers whose delivery can be feasibly reduced, then decreases the amount delivered as much as possible;
- *Random insertion*: selects a route and one of the customers not served in the period of the route and then inserts the largest feasible delivery to visit the customer;
- *Random split*: chooses a route and one of its visits, then transfers half of the delivery to another time period.

Each call to the perturbation algorithm activates a single operator, which can perturb up to `max_perturb` elements of the solution. The operators are called at random until one of them alters at least one element of the solution. Similar to the local search phase, only feasible solutions are allowed. Because these operators perturb different parts of the incumbent solution in a random manner, it permits the algorithm to avoid local optimal solutions. Finally, note that `max_perturb` is the only parameter that has to be set by the user in the proposed ILS.

3.4 Simulated annealing metaheuristic

SA is a metaheuristic that avoids getting trapped in local optima of the search space by probabilistically accepting solutions that temporarily produce degradations in the current search point (Kirkpatrick et al., 1983). In a SA algorithm, given an initial solution, at each iteration a random neighbor of the current solution is selected. If the new solution is better than the current one, the process continues from this new solution. On the other hand, if the new solution is worse than the current one, it is accepted under a given probability that depends on the change of the objective function and the current temperature of the process. As noted by Galvão et al. (2005), SA is a metaheuristic that can be easily implemented and is often less time-consuming than more sophisticated metaheuristics.

We have develop a SA that combines the search strategy of a basic simulated annealing algorithm with additional features such as multi-start search, an intensification phase and multiple iterations per temperature. The pseudo-code of the proposed method is shown in Algorithm 3.4. It starts with an initial solution generated by the construction heuristic described in Section 3.2 (line 2). Similar to the ILS algorithm, if the construction heuristic cannot find a feasible solution, the SA algorithm terminates; otherwise, the search process begins. This initial solution is used to define the `n_sol` multi-start points (line 5) that will be perturbed in the main loop of the algorithm. At each inner iteration from 1 to `maxI`, the algorithm picks a random neighbor of each of the current solutions S_i (line 10), for $i = 1, \dots,$ `n_sol`. If the resulting neighbor $S_i^{'}$ is better than the current solution S_i, then it becomes the new current solution (line 12). Otherwise, a real number $r \in [0, 1]$ is chosen at random (line 14), and the neighbor replaces the current solution only if the random number is less than the computed probability (line 15).

After running `maxI` inner iterations with the same temperature, an intensification phase is performed applying the RVND heuristic (see Section 3.3.1) for each of the `n_sol` resulting solutions (line 19-22). If a new best solution is found, it is stored. Next, the best solution is defined as the starting point for all the `n_sol` multi-start solutions (line 23), and the temperature is decreased (line 24). The method terminates after reaching at least one of the two stopping criteria: minimum temperature and running time. The proposed SA uses similar features to the algorithm of Yu and Lin (2014), who developed a SA metaheuristic for the location-routing problem with simultaneous pickup and delivery.

Algorithm 3.4: Simulated annealing

```
1  begin
2      S_0 ← Construct initial solution;
3      if S_0 ≠ ∅ then
4          S* ← S_0;
5          for i = 1, ..., n_sol do  S_i ← S_0 ;
6          Temp ← Temp_0;
7          while stop criterion is not met do
8              for I = 1, ..., maxI do
9                  for i = 1, ..., n_sol do
10                     S_i' ← randomNeighbor(S_i);
11                     Δ ← f(S_i') − f(S_i);
12                     if Δ < 0 then  S_i ← S_i' ;
13                     else
14                         r ~ U[0, 1];
15                         if r < exp(−Δ/Temp) then  S_i ← S_i' ;
16                     end
17                 end
18             end
19             for i = 1, ..., n_sol do
20                 S_i ← multi-RVND(S_i);
21                 if f(S_i) < f(S*) then  S* ← S_i ;
22             end
23             for i = 1, ..., n_sol do  S_i ← S* ;
24             Temp ← αTemp;
25         end
26     end
27 end
```

3.4.1 Neighborhood setting and parameters of the algorithm

In a SA metaheuristic, the neighborhood must be defined allowing the algorithm to reach the optimal solution. This involves designing of operators which must include as many solution attributes values as possible. Thus, in our implementation we use all local search operators of the RVND heuristic, described in Section 3.3.1. Each time SA tries a movement to a neighboring solution (line 10 of Algorithm 3.4), the algorithm chooses one of these operators at random and applies a single random movement. In this phase, all operators have the same probability of being selected. As they alter distinct characteristics of the solutions, they help to explore a large scope of the search space.

On the other hand, different to ILS, our SA has many parameters and all of them influence the behavior of the algorithm. In total, the proposed SA algorithm uses five parameters: $\texttt{Temp}_0$, $\texttt{Temp}_{\texttt{min}}$, α, `n_sol` and `maxI`, where $\texttt{Temp}_0$ represents the initial temperature, $\texttt{Temp}_{\texttt{min}}$ is the minimum temperature (potential stop criterion), α is the cooling rate of the algorithm, `n_sol` defines the number of multi-start points of the metaheuristic and `maxI` is the number of iterations performed at a particular temperature.

3.5 Remarks on the computational implementation

In this section, we briefly describe some additional implementation challenges that we addressed during the development of the metaheuristic algorithms and how we overcame them. Please notice that some other implementation details were already described during the description of the components of the metaheuristic. First of all, in the construction heuristic the simple decomposition of the decisions was not enough to find reasonably good feasible solutions. Thus, we tried to include some degree of integration in the time dimension by incorporating an anticipation strategy with the dynamic parameter `look_ahead`. In addition, the parameter `ratio_demand` also helped to better determine the delivery decisions in these solutions. Second of all, the straightforward implementation of the metaheuristics did not provide good solutions at first, so we had to add additional features to them. For the SA algorithm, the multi-start strategy helped to define different search trajectories simultaneously. Even though we use the same solution for each one of the multiple starting points, the randomized nature of the algorithm (probably) defines different search patterns and helps the algorithm to have a better performance (compared to a SA with only a single initial solution). Additionally, the intensification phase included in this metaheuristic was a key point to achieve the desired performance. For the ILS, using the multi-start RVND as local search significantly improved the performance of the algorithm (compared to an ILS with VND as local search) as this component provides a strong and balanced intensification phase for the metaheuristic.

Finally, given that we do not use any mathematical programming component within the metaheuristics, we had to design specific local search operators to handle the visit and delivery decisions on the solutions (see Section 3.2.1). Nevertheless, the results presented in Section 4 indicate that the developed operators were able to effectively handle these decisions, as we outperform heuristic methods that use operators based on mathematical programming.

3.6 Computational experiments

In this section, we present the computational experiments using the proposed ILS and SA to solve the IRP. All the algorithms were implemented in C++ and the experiments performed on a Linux PC with an Intel Core i7-2600 3.4 GHz processor and 16 GB of RAM. It is worth mentioning that it is not possible to solve formulation (2.1)-(2.14) directly in a general-purpose optimization software given the presence of the SECs (2.8), whose size grows exponentially with an increasing number of customers. Thus, it is necessary to add them dynamically when violated in a branch-and-cut algorithm. However, even with this type of algorithm only small-sized instances can be solved to optimality within reasonably running times, requiring the use of more specialized branch-and-cut methods (Archetti et al., 2007; Coelho and Laporte, 2014a; Avella et al., 2018).

3.6.1 Problem instances

Both metaheuristics were tested on two sets of benchmark instances of Archetti et al. (2007) and Archetti et al. (2012), respectively. Both sets were proposed for the single-vehicle basic variant of the IRP. The first set is composed of 160 instances with up to 50 customers and six time periods. The instances are further divided into four sets: H3-I, L3-I, H6-I and L6-I. The second set has 60 problem instances with six time periods and up to 200 customers. They are divided into two sets: H6-II and L6-II. In all sets, H (L) stands for high (low) inventory holding costs when compared to the travel costs, while the digit (3 or 6) indicates the number of time periods in all instances of the set. The product consumption of each customer and the quantity available at the supplier in each time period are constant through the planning horizon. Unitary holding costs are provided and travel costs correspond to Euclidean distances rounded to the nearest integer. Following the common practice in the literature (Coelho and Laporte, 2013b; Desaulniers et al., 2016; Archetti et al., 2017a; Avella et al., 2018), for the multi-vehicle we consider a homogeneous fleet of vehicles with size ranging from two to five. The original vehicle capacity is divided by the number of available vehicles and then rounded to the nearest integer when the objective function is the total cost (OF1) and, following Archetti et al. (2017b), rounded to the nearest lower integer when the objective function is the logistic ratio (OF2). In the case with five vehicles, two instances (one of set H6-I and one of set L6-I) are infeasible because the capacity of a single vehicle is not enough to serve the daily consumption of one of the customers (and no splitting is allowed). These two instances were discarded. Therefore, in total we have 1098 problem instances to test our algorithms.

3.6.2 Tuning the parameters of the algorithms

To determine an appropriate combination for the parameters of the metaheuristics, we used the ParamILS algorithm of Hutter et al. (2009). ParamILS is an automated local search approach that explores the parameter configuration space and can be used for either minimizing running time in decision problems or for maximizing solution quality in optimization problems. ParamILS requires a discrete range for each parameter and starts with one parameter configuration defined by the user. Thereafter, the algorithm assesses a sequence of combinations applying local searches until reaching a quality threshold or a time limit. In our implementation, we provided the tool with the 798 benchmark instances of Archetti et al. (2007) and it randomly chooses a subset out of these instances to perform the calibration. We defined a cut-off time of two hours for the parameter tuning of each metaheuristic, limiting the running time of each single problem instance to five seconds.

Table 3.1 shows the parameters of the metaheuristics, the values returned in the best configuration of ParamILS and the tested ranges. These ranges were defined through preliminary experiments. For ILS, the value for the perturbation parameter defines a suitable strength of the disturbance in each iteration of the algorithm. For SA, note that the performance of the algorithm depends on the effort before each temperature decrease – which is defined by `n_sol` and `maxI` – and on the overall number of iterations, which depends on α, $\texttt{Temp}_0$ and $\texttt{Temp}_{\texttt{min}}$. Thus, as expected, ParamILS returned values that extend the search process of SA. It is worth mentioning, however, that similar configurations were also derived through preliminary empirical tests. In addition, we set the number of iterations of the multi-start RVND to five, which provides an adequate balance between solution quality and computational effort, according to prior tests. These configurations were adopted in all the subsequent experiments.

Metaheuristic	**Parameter**	**Value**	**Tested range**
ILS	`max_perturb`	7	{1, 3, ..., 15}
SA	`n_sol`	10	{1, 2, ..., 10}
	`maxI`	25	{1, 5, 10, ..., 30}
	$\texttt{Temp}_0$	3000	{500, 1000, ..., 3500}
	$\texttt{Temp}_{\texttt{min}}$	10^{-6}	{1, 0.1, ..., 10^{-6}}
	α	0.99	{0.95, 0.96, ..., 0.99}

Table 3.1: Parameters of the metaheuristics

3.6.3 Performance comparison using small instances

In the first computational experiment, we compare the performance of the proposed metaheuristics according to different running time limits when minimizing OF1. As both metaheuristics have random components, each instance was run five times with a time limit of 5 and 30 seconds. It is worth mentioning that when longer running times were tested, no significant gains were achieved compared to the additional computational effort. We also carried out tests using improved and/or different solutions as starting points for both metaheuristics, but no significant improvement was attained. Therefore, the construction heuristic of Section 3.2 was used as the source of the initial solutions for the metaheuristics in all the computational experiments.

Tables 3.2 and 3.3 summarize the best out of five results within 5 and 30 seconds, respectively. The first three columns in the tables give the name of the instance set, the number of available vehicles and the total number of instances in the group, respectively. We grouped the instances of each set according to the number of vehicles, so that each row in the tables represents the average over all instances in the corresponding group (same set and same number of vehicles).

For each method, the tables show the value of the objective function of the solutions (total cost $\bar{z}$); the time required to find the final solution (time to $\bar{z}$); the optimality gap (opt gap), computed using the formula $(\bar{z} - \underline{z})/\underline{z}$, where $\bar{z}$ is the value of the objective function of the solution found by the metaheuristics and $\underline{z}$ is the best lower bound (LB) reported by the BPC algorithm of Desaulniers et al. (2016) and the B&C of Coelho and Laporte (2014a) (available online at `http://www.leandro-coelho.com`); and the relative difference between the results obtained by the method and the best upper bounds (UB) reported in the literature (best UB gap), computed using the formula $(\bar{z} - z^{best})/z^{best}$, where z^{best} is the best known solution from the literature. The best UBs correspond to the solutions provided by the Adaptive Large Neighborhood Search (ALNS)-based hybrid method of Adulyasak et al. (2014b), the B&C of Adulyasak et al. (2014a) and the previously mentioned exact methods. The best average total cost in each row is emphasized in boldface and the average of each column is shown in the last row of the tables (avg). Additionally, the values of the average optimality gaps of the solutions for five-seconds time limit are shown in Figures 3.1 and 3.2.

Set	nV	# of instances	Iterated Local Search				Simulated Annealing			
			total cost $\bar{z}$	time to $\bar{z}$	opt gap*	best UB gap**	total cost $\bar{z}$	time to $\bar{z}$	opt gap*	best UB gap**
H3-I	1	50	9279.03	0.40	0.43%	0.43%	**9243.77**	0.83	0.09%	0.09%
	2	50	**9625.80**	0.84	0.54%	0.54%	9628.56	1.85	0.41%	0.41%
	3	50	10081.15	1.01	0.66%	0.59%	**10081.01**	2.71	0.55%	0.47%
	4	50	10569.55	1.44	0.90%	0.54%	**10558.53**	2.45	0.69%	0.33%
	5	50	11012.82	1.76	1.10%	0.03%	**10999.55**	2.72	0.93%	-0.14%
L3-I	1	50	3026.58	0.67	1.62%	1.62%	**2973.88**	0.89	0.13%	0.13%
	2	50	3420.15	1.06	3.20%	3.18%	**3359.39**	2.03	1.20%	1.18%
	3	50	3865.27	1.25	3.14%	2.80%	**3813.72**	2.39	1.56%	1.21%
	4	50	4315.54	1.54	2.51%	1.35%	**4287.42**	2.68	1.78%	0.61%
	5	50	4767.22	1.51	3.33%	0.63%	**4737.48**	2.43	2.44%	-0.29%
H6-I	1	30	**13001.29**	2.15	0.71%	0.71%	13027.87	2.85	0.82%	0.82%
	2	30	**14071.98**	2.90	1.72%	1.65%	14092.38	3.20	1.77%	1.71%
	3	30	**15299.94**	3.07	2.34%	1.67%	15322.64	3.90	2.38%	1.70%
	4	30	**16617.15**	3.35	2.99%	-0.07%	16618.39	3.59	2.79%	-0.28%
	5	29	**18326.07**	3.17	3.71%	-0.53%	18347.29	4.01	3.95%	-0.29%
L6-I	1	30	**5902.19**	2.33	1.91%	1.91%	5922.08	2.38	2.11%	2.11%
	2	30	**6915.65**	3.01	3.12%	2.88%	6937.58	2.79	3.31%	3.06%
	3	30	**8189.08**	2.84	4.65%	3.14%	8225.44	3.12	5.00%	3.48%
	4	30	**9476.69**	2.98	5.09%	0.53%	9492.78	3.54	5.25%	0.66%
	5	29	**11003.49**	2.89	6.23%	-3.18%	11026.92	3.50	6.46%	-3.03%
avg			9438.33	2.01	2.50%	1.02%	9434.83	2.69	2.18%	0.70%

* Best LB from Desaulniers et al. (2016) and Coelho and Laporte (2014a);
** Best UB from Desaulniers et al. (2016) (*BPC*), Coelho and Laporte (2014a) (*B&C*) and Adulyasak et al. (2014a,b) (*B&C* & ALNS).

Table 3.2: Best results with the metaheuristics for the 5 seconds time limit

The results of the experiment with a time limit equal to 5 seconds (Table 3.2) show that both metaheuristics have a similar overall performance in terms of the average total cost of the best obtained solutions. Notice that when considering the best out of five solutions, SA found better results for all but one instance group ($nV = 2$) with three time periods (H3 and L3), whereas ILS found, on average, slightly better total costs for all instance sets with a longer planning horizons (H6 and L6). This fact can be explained by the strength of the diversification, which is stronger in ILS than in SA due to the larger number of elements modified in each iteration. It can be noted that the time to find the final solution (time to $\bar{z}$) is very short and, as expected, tends to increase with the number of vehicles and time periods. The average optimality gaps of the solutions also tend to increase with the number of vehicles and periods. Nevertheless, this conclusion must be taken cautiously, as these LBs were provided by two different exact solution methods and many of these bounds may not correspond to optimal solutions. Finally, negative average gaps to the best UBs indicate that, on average, the solutions found are better than the best solutions reported in the literature.

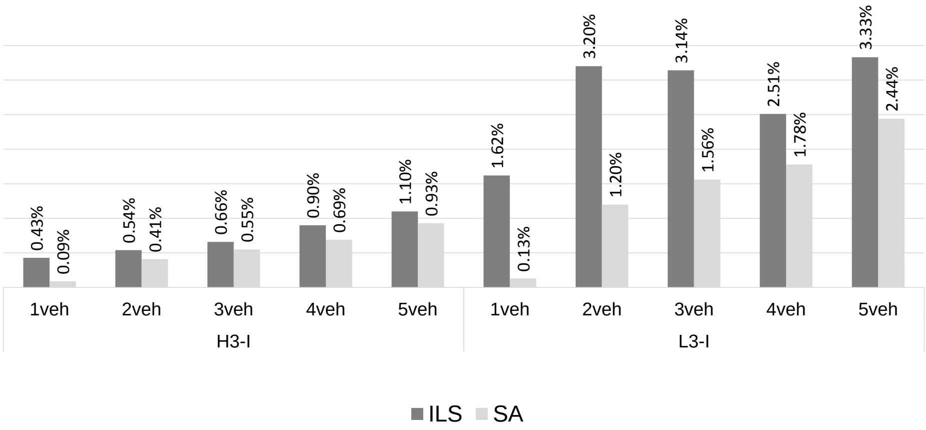


Figure 3.1: Optimality gaps of the solutions found by the metaheuristics for the instances with three time periods

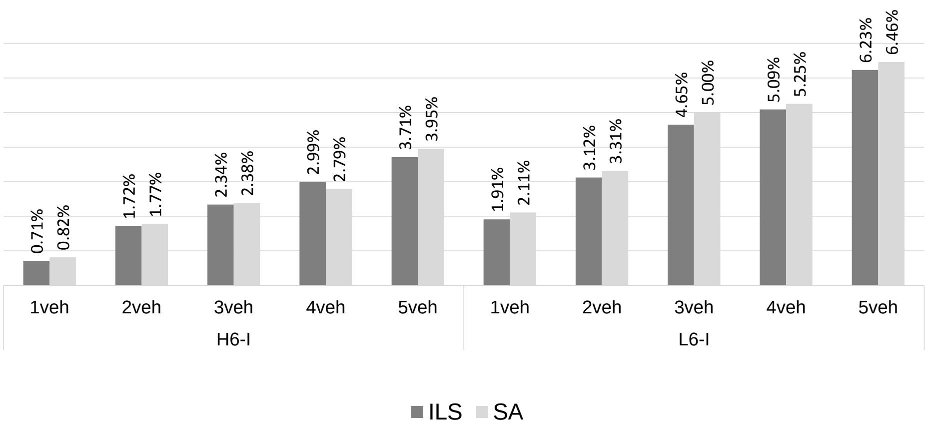


Figure 3.2: Optimality gaps of the solutions found by the metaheuristics for the instances with six time periods

A similar behavior is observed in the experiment with a longer running time (Table 3.3), where SA was slightly superior to ILS in terms of the overall average results. This is mainly because of the better performance of SA on the instances with shorter planning horizon. Notice that the time instant in which ILS and SA find the final solutions (time to $\bar{z}$) is on average one third of the maximum running time. This highlights the ability of the proposed metaheuristic algorithms to find good feasible solutions in relatively short running times. Finally, it can be observed that the optimality gaps are quite low considering the relatively short running times.

Additionally, we obtain negative average gaps to the best UBs indicating the finding of better solutions compared to the best solutions from the literature. Specifically, ILS and SA found 65 and 64 new best known solutions for these sets, respectively. Detailed results on all instances are available online at `http://www.dep.ufscar.br/docentes/munari/irp`.

Set	nV	# of instances	Iterated Local Search				Simulated Annealing			
			total cost $\bar{z}$	time to $\bar{z}$	opt gap*	best UB gap**	total cost $\bar{z}$	time to $\bar{z}$	opt gap*	best UB gap**
H3-I	1	50	9280.45	2.07	0.37%	0.37%	**9236.51**	1.43	0.04%	0.04%
	2	50	9612.83	5.35	0.33%	0.33%	**9601.19**	4.34	0.19%	0.19%
	3	50	10060.74	5.17	0.47%	0.40%	**10046.70**	5.61	0.30%	0.22%
	4	50	10538.93	7.15	0.67%	0.32%	**10515.64**	7.94	0.39%	0.03%
	5	50	10993.96	7.27	0.97%	-0.10%	**10961.50**	8.03	0.67%	-0.40%
L3-I	1	50	3019.71	3.11	1.41%	1.41%	**2972.25**	1.50	0.09%	0.09%
	2	50	3393.57	6.57	2.32%	2.31%	**3339.63**	3.82	0.75%	0.74%
	3	50	3850.93	5.62	2.82%	2.48%	**3793.52**	5.00	1.11%	0.76%
	4	50	4302.44	6.43	2.24%	1.10%	**4253.22**	6.73	1.10%	-0.04%
	5	50	4734.09	8.52	2.69%	0.03%	**4690.57**	6.58	1.65%	-1.00%
H6-I	1	30	**12999.55**	11.21	0.72%	0.72%	13000.34	9.67	0.66%	0.66%
	2	30	**14005.35**	15.23	1.24%	1.17%	14013.43	15.18	1.27%	1.21%
	3	30	**15256.78**	17.02	2.18%	1.51%	15273.90	16.16	2.09%	1.41%
	4	30	16539.82	14.06	2.50%	-0.55%	**16519.03**	16.69	2.24%	-0.80%
	5	29	18219.63	18.22	3.19%	-1.02%	**18205.78**	18.37	3.17%	-1.04%
L6-I	1	30	**5898.11**	6.71	1.72%	1.72%	5902.30	9.67	1.70%	1.70%
	2	30	**6856.92**	15.24	2.32%	2.07%	6926.01	13.74	3.20%	2.96%
	3	30	**8112.11**	15.14	3.79%	2.30%	8163.76	14.60	4.33%	2.83%
	4	30	9400.22	17.99	4.32%	-0.19%	**9375.24**	17.16	4.07%	-0.46%
	5	29	10936.91	13.68	5.66%	-3.70%	**10901.84**	18.11	5.33%	-4.00%
avg			9400.65	10.09	2.10%	0.63%	9384.62	10.02	1.72%	0.26%

* Best LB from Desaulniers et al. (2016) and Coelho and Laporte (2014a);
** Best UB from Desaulniers et al. (2016) (*BPC*), Coelho and Laporte (2014a) (*B&C*) and Adulyasak et al. (2014a,b) (*B&C* & ALNS).

Table 3.3: Best results with the metaheuristics for the 30 seconds time limit

Furthermore, we applied a Wilcoxon sum rank test (Wilcoxon, 1945) to verify the significance of the difference in the results for instances with different number of time periods. To compare the methods, we used the gap to the best LB of each instance for the results with 30 seconds of time limit (considering all instance sets and vehicles). The results with a confidence level of 95% show that for instances with three time periods (sets H3 and L3) there exists a statistically significant difference of the results, confirming the advantages of SA on these sets. For the instances with six time periods (H6 and L6), no statistically significant difference was identified.

Regarding the characteristics of the instances, notice that when the number of vehicles increases for the same instance set, the average total cost also increases as a result of larger total travel costs. The average time to find the final solution of the algorithms also increases with the number of vehicles as a consequence of the enlargement of the search space.

Tables 3.4 and 3.5 present the average, best and worst results of the five executions of ILS and SA for 5 and 30 seconds, respectively. As expected, for both metaheuristics the average variability (stated as the range of the results) decreased when the maximum running time increased, as the metaheuristics found better results in each run. However, notice that for both time limits, the average variability of ILS was greater than the variability of SA as a result of the larger diversification of ILS when compared to SA.

Set	nV	# of instances	ILS: total cost $\bar{z}$			SA: total cost $\bar{z}$		
			best	average	worst	best	average	worst
H3-I	1	50	9279.03	9309.74	9356.75	9243.77	9249.57	9260.28
	2	50	9625.80	9701.39	9808.67	9628.56	9658.36	9700.25
	3	50	10081.15	10175.20	10291.94	10081.01	10131.18	10211.43
	4	50	10569.55	10635.82	10718.98	10558.53	10616.81	10679.63
	5	50	11012.82	11097.49	11203.15	10999.55	11057.32	11121.41
L3-I	1	50	3026.58	3050.97	3080.86	2973.88	2983.12	2990.23
	2	50	3420.15	3475.33	3553.43	3359.39	3384.62	3421.43
	3	50	3865.27	3958.06	4073.30	3813.72	3857.83	3918.40
	4	50	4315.54	4405.08	4503.67	4287.42	4345.56	4418.41
	5	50	4767.22	4838.59	4936.19	4737.48	4795.66	4863.58
H6-I	1	30	13001.29	13063.83	13130.96	13027.87	13110.28	13214.61
	2	30	14071.98	14193.66	14328.40	14092.38	14224.19	14354.38
	3	30	15299.94	15438.99	15580.10	15322.64	15470.71	15624.82
	4	30	16617.15	16791.20	16988.53	16618.39	16772.60	16938.33
	5	29	18326.07	18517.54	18701.04	18347.29	18508.58	18672.77
L6-I	1	30	5902.19	5959.66	6047.75	5922.08	6013.46	6099.26
	2	30	6915.65	7042.84	7193.96	6937.58	7075.18	7236.74
	3	30	8189.08	8307.91	8471.88	8225.44	8345.42	8491.93
	4	30	9476.69	9661.32	9849.79	9492.78	9648.85	9810.18
	5	29	11003.49	11217.33	11440.19	11026.92	11207.47	11398.36
avg			9438.33	9542.10	9662.98	9434.83	9522.84	9621.32

Table 3.4: Best, average and worst results with the metaheuristics for the 5 seconds time limit

Set	nV	# of instances	ILS: total cost $\bar{z}$			SA: total cost $\bar{z}$		
			best	average	worst	best	average	worst
H3-I	1	50	9280.45	9303.69	9321.97	9236.51	9242.46	9252.88
	2	50	9612.83	9674.20	9766.45	9601.19	9622.87	9655.45
	3	50	10060.74	10150.98	10281.27	10046.70	10086.24	10132.12
	4	50	10538.93	10614.92	10708.50	10515.64	10560.23	10622.26
	5	50	10993.96	11065.18	11159.62	10961.50	11008.50	11061.95
L3-I	1	50	3019.71	3034.47	3061.41	2972.25	2976.80	2979.49
	2	50	3393.57	3447.69	3519.30	3339.63	3353.72	3374.29
	3	50	3850.93	3927.49	4032.13	3793.52	3824.89	3867.02
	4	50	4302.44	4374.89	4454.13	4253.22	4299.63	4364.31
	5	50	4734.09	4807.41	4904.22	4690.57	4741.18	4789.72
H6-I	1	30	12999.55	13045.71	13103.85	13000.34	13101.77	13213.67
	2	30	14005.35	14110.07	14230.86	14013.43	14157.04	14315.04
	3	30	15256.78	15352.82	15471.61	15273.90	15389.39	15498.65
	4	30	16539.82	16682.64	16803.69	16519.03	16656.39	16795.07
	5	29	18219.63	18395.72	18582.95	18205.78	18355.10	18498.01
L6-I	1	30	5898.11	5941.87	5997.64	5902.30	5997.49	6107.17
	2	30	6856.92	6976.22	7139.05	6926.01	7030.61	7144.63
	3	30	8112.11	8222.69	8355.93	8163.76	8277.60	8399.82
	4	30	9400.22	9551.14	9738.30	9375.24	9515.40	9656.19
	5	29	10936.91	11105.22	11279.32	10901.84	11065.31	11205.51
avg			9400.65	9489.25	9595.61	9384.62	9463.13	9546.66

Table 3.5: Best, average and worst results with the metaheuristics for the 30 seconds time limit

To further analyze the performance of ILS and SA regarding different characteristics of the instances, Table 3.6 shows the detailed results (considering the average of the five executions of each method for the 30 seconds time limit) for the instance sets H3 and L6 with one and five vehicles, respectively. These instances represent extreme features of all tested instances. It is worth highlighting that all solutions for the instances with one single vehicle were solved to optimality by exact methods (Desaulniers et al., 2016; Coelho and Laporte, 2014a). The results are separated according to the number of customers (nC) and the columns maintain the same meaning used in Tables 3.2 and 3.3. Notice that the average time required to find the obtained solutions (time to $\bar{z}$) increases with the number of customers in the instances. Additionally, the optimality gaps of SA are slightly lower than the gaps of ILS for the single-vehicle case. For the five-vehicle instances the average optimality gaps and relative differences to the best reported solutions are similar for both metaheuristics. In particular, for the instances with 25 to 30 customers in set L6, which are the largest ones, the average cost of SA and ILS were better than the best UBs from the literature (differences of more than 14%).

Set	nV	nC	# of instances	Iterated Local Search				Simulated Annealing			
				total cost $\bar{z}$	time to $\bar{z}$	opt gap	best UB gap	total cost $\bar{z}$	time to $\bar{z}$	opt gap	best UB gap
H3-I	1	5-10	10	3268.94	0.01	0.00%	0.00%	3268.94	0.02	0.00%	0.00%
		15-20	10	6336.65	1.12	0.08%	0.08%	6330.75	0.18	0.00%	0.00%
		25-30	10	9973.39	3.07	0.23%	0.23%	9952.32	0.27	0.02%	0.02%
		35-40	10	12050.18	2.99	0.63%	0.63%	11992.48	1.60	0.14%	0.14%
		45-50	10	14773.09	3.17	0.94%	0.94%	14638.08	5.07	0.01%	0.01%
L6-I	5	5-10	9	8823.59	5.30	3.80%	2.79%	8765.18	10.56	3.05%	2.05%
		15-20	10	11026.73	16.02	5.78%	1.10%	11052.16	16.94	6.05%	1.35%
		25-30	10	12749.08	18.89	7.21%	-14.34%	12674.51	26.07	6.66%	-14.78%

Table 3.6: Results for the H3-I instances with one vehicle and L6-I instances with five vehicles

3.6.4 Performance comparison using larger instances

In this section we analyze the performance of the metaheuristics when applied to larger problem instances (H6-II and L6-II). In the literature, these instances were addressed in the single-vehicle case by Archetti et al. (2012) (with a hybrid heuristic) and in the multi-vehicle case by Coelho et al. (2012) (with a hybrid heuristic) and Coelho and Laporte (2013b) (with a B&C algorithm). As these instances are considerably larger, we run each metaheuristic five times with 60 seconds as stop criterion. Table 3.7 summarizes the best out of five results obtained for OF1. The table columns maintain the same meaning of the previous tables. The results show that both ILS and SA have similar average performance (the results of SA are only 0.68% greater than the results of ILS) and that the time to reach the final solution is near to the time limit. The latter fact exposes the difficulty of solving these larger problem instances, as the algorithms required almost all the available time to reach the final solution. The gap to the best UBs from the literature is about -15% for both metaheuristics, considering all instances. This shows that the metaheuristics are also effective to solve these large problem instances. Specifically, ILS and SA found 224 and 219 new best known solutions for these sets, respectively.

3.6.5 Comparison with prior results

This section compares the performance of ILS and SA to the state-of-the-art heuristic methods from the literature when minimizing OF1. In this sense, for the small instances (H3-I, L3-I, H6-I and L6-I) a comparison is only possible with the ALNS-based hybrid method of Adulyasak

Set	nV	# of instances	Iterated Local Search		Simulated Annealing	
			total cost $\bar{z}$	time to $\bar{z}$	total cost $\bar{z}$	time to $\bar{z}$
H6-II	1	30	65500.43	51.84	65643.04	57.45
	2	30	66483.60	52.86	66922.70	56.16
	3	30	67763.71	53.05	68182.02	57.71
	4	30	69488.33	55.51	69789.13	57.12
	5	30	71360.96	55.08	71602.43	57.54
L6-II	1	30	17386.48	52.45	17390.49	55.21
	2	30	18284.16	54.43	18735.70	56.08
	3	30	19552.16	54.71	19921.95	56.67
	4	30	21315.93	55.81	21711.74	56.94
	5	30	23206.84	54.47	23439.87	58.33
avg			44034.26	54.02	44333.91	56.92

Table 3.7: Best results with the larger instances for the 60 seconds time limit

et al. (2014b), as it is the only heuristic with publicly available results (partly, though). Table 3.8 reports the average relative differences between the results found by ILS and SA (best out of five runs within 30 seconds) and the results of Adulyasak et al. (2014b). Column nC displays the range of customers considered in the instances tested. The results are grouped according to the number of vehicles. It can be observed that both metaheuristics proposed in this chapter outperform the ALNS-based hybrid method of Adulyasak et al. (2014b) on average, as only negative relative differences are presented for the instance sets. Unfilled cells are due to incomplete report of the results of the ALNS-based hybrid method. In total, ILS and SA found, respectively, 65 and 64 new best known solutions when compared to all publicly available feasible solutions (Desaulniers et al., 2016; Coelho and Laporte, 2014a; Adulyasak et al., 2014b), most of them for instances with four and five vehicles.

Set	nC	ILS			SA		
		2 veh	3 veh	4 veh	2 veh	3 veh	4 veh
H3-I	5-50	-1.88%	-3.29%	-3.00%	-2.07%	-3.46%	-3.22%
L3-I	5-50	-1.75%	-3.99%	-6.83%	-3.31%	-5.51%	-7.76%
H6-I	5-25	-1.57%	-0.66%	–	-1.60%	-0.90%	–
L6-I	5-25	-3.03%	-1.79%	–	-2.35%	-1.41%	–
avg		-2.06%	-2.43%	-4.91%	-2.33%	-2.82%	-5.49%

Table 3.8: Relative differences to the ALNS of Adulyasak et al. (2014b)

On the other hand, for the larger problem instances, when comparing to the hybrid heuristic algorithm of Archetti et al. (2012) considering instances with only one vehicle (the only case addressed in their paper), ILS and SA found solutions with average objective value 2.86% and 2.88% greater than their results, respectively. Comparing to the approach of Coelho et al. (2012) for the multi-vehicle case (2 to 5 vehicles), solutions with average cost 20% smaller were found by both algorithms, on average.

3.6.6 Solving the IRP with the logistic ratio

In this section, we show the results of the proposed metaheuristics when used to solve the IRP with the logistic ratio as the objective function (OF2). Studying the logistic ratio in the context of the IRP may be valuable in practice because, as pointed out by Archetti et al. (2017b), the logistic ratio absorbs the long-term impact of a short-term planning, as it focuses on the cost efficiency of the distribution process. Minor changes were necessary to adapt the metaheuristics to address this alternative criterion. This easy adaptation is a desirable feature in practice, especially if in addition it maintains a similar computational performance of the method.

Tables 3.9 and 3.10 show a comparison between the results of ILS and SA when minimizing OF1 and OF2, respectively. They present the best results out of five runs for each objective function within a time limit of 30 seconds. We grouped the instances according to their respective instance set and number of vehicles, as in Tables 3.2 and 3.3. The first three columns have the same meaning as in those tables. Columns 4-7 show the results of the metaheuristics when minimizing the logistic ratio. Column 4 displays the cost of the solution calculated as OF1 (in these experiments we are minimizing OF2); column 5 shows the total routing cost of the solution, which corresponds to the numerator of OF2; column 6 displays the quantity delivered by all routes in the solution, which corresponds to the denominator of OF2; and column 6 represents the value of the logistic ratio (OF2). Columns 8-11 display the relative difference between the values of columns 4-7 and the results when minimizing the total cost. Each row shows the average of the results over all instances in the group. In addition, Figure 3.3 presents the relative differences of the results when minimizing the logistic ratio compared to the results when minimizing the total cost, for the ILS metaheuristic.

Set	nV	# of instances	minimizing logistic ratio				difference to minimizing total cost $\bar{z}$			
			total cost $\bar{z}$	routing cost	quantity delivered	logistic ratio	total cost $\bar{z}$	routing cost	quantity delivered	logistic ratio
H3-I	1	50	9949.26	2866.22	3738.54	1.03	7.21%	23.37%	62.90%	-23.70%
	2	50	10543.53	3363.62	3758.78	1.26	9.68%	26.46%	59.71%	-21.40%
	3	50	11180.78	3945.84	3718.33	1.54	11.13%	26.93%	54.17%	-21.04%
	4	50	11773.42	4580.52	3669.32	1.82	11.71%	27.36%	54.53%	-20.13%
	5	50	12432.80	5254.58	3645.04	2.12	13.09%	29.72%	55.79%	-17.25%
L3-I	1	50	3672.20	2848.10	3736.94	1.03	21.61%	22.59%	63.98%	-24.46%
	2	50	4290.49	3337.26	3790.18	1.26	26.43%	23.70%	59.11%	-20.95%
	3	50	4928.19	3969.82	3741.62	1.54	27.97%	25.82%	55.66%	-20.61%
	4	50	5517.74	4665.82	3679.99	1.83	28.25%	29.36%	56.95%	-20.09%
	5	50	6111.39	5231.68	3605.62	2.12	29.09%	29.55%	55.85%	-18.93%
H6-I	1	29	13691.95	5571.93	5363.17	1.24	5.33%	9.18%	22.14%	-11.02%
	2	30	14973.04	6844.40	5289.17	1.57	6.91%	12.34%	21.04%	-9.96%
	3	30	16363.90	8322.27	5226.50	1.94	7.26%	13.21%	19.88%	-9.96%
	4	30	17893.91	9646.53	5235.86	2.34	8.19%	11.93%	20.31%	-8.77%
	5	30	19622.60	11303.45	5305.48	2.64	7.70%	11.76%	19.02%	-8.19%
L6-I	1	29	6492.88	5591.63	5342.03	1.23	10.08%	9.50%	22.43%	-12.19%
	2	30	7711.30	6770.97	5287.93	1.56	12.46%	11.67%	20.72%	-9.90%
	3	30	9148.44	8152.33	5249.50	1.94	12.78%	11.41%	21.17%	-9.44%
	4	30	10671.31	9673.17	5206.60	2.34	13.52%	12.40%	20.15%	-8.58%
	5	30	12346.33	11260.07	5315.31	2.65	12.89%	11.22%	19.09%	-8.10%
avg			10465.77	6160.01	4495.30	1.75	14.16%	18.97%	39.23%	-15.23%

Table 3.9: Results of the ILS minimizing logistic ratio vs total cost

It can be observed that the average total cost and routing cost increased when minimizing the logistic ratio because of the substantial increase of the average quantities delivered (approximately 39% for both metaheuristics). As a result, an average reduction of 15.23% and 15.75% in the logistic ratio was obtained for the ILS and SA when compared to the traditional objective function, respectively. Observe that lower logistic ratios are obtained for instances in the

Set	nV	# of instances	minimizing logistic ratio				difference to minimizing total cost $\bar{z}$			
			total cost $\bar{z}$	routing cost	quantity delivered	logistic ratio	total cost $\bar{z}$	routing cost	quantity delivered	logistic ratio
H3-I	1	50	9822.32	2791.94	3658.04	1.02	6.34%	22.52%	59.99%	-23.74%
	2	50	10350.74	3323.30	3654.20	1.25	7.81%	25.37%	54.18%	-21.36%
	3	50	10995.37	3907.88	3635.48	1.53	9.44%	26.16%	52.15%	-21.86%
	4	50	11729.78	4618.90	3682.04	1.81	11.55%	29.48%	55.99%	-20.15%
	5	50	12362.85	5228.20	3656.70	2.10	12.78%	30.36%	56.74%	-18.65%
L3-I	1	50	3533.94	2786.72	3669.32	1.02	18.90%	22.44%	60.67%	-23.90%
	2	50	4094.53	3273.84	3696.60	1.24	22.60%	23.84%	61.03%	-22.06%
	3	50	4673.16	3928.16	3631.79	1.53	23.19%	26.81%	53.91%	-21.78%
	4	50	5423.38	4609.62	3684.03	1.81	27.51%	29.60%	59.06%	-20.87%
	5	50	5993.73	5236.08	3586.34	2.10	27.78%	31.09%	56.46%	-19.24%
H6-I	1	29	13643.48	5577.43	5320.43	1.24	4.95%	9.44%	21.18%	-12.40%
	2	30	14856.60	6804.73	5233.80	1.56	6.02%	11.58%	19.04%	-10.32%
	3	30	16296.64	8234.43	5232.57	1.94	6.70%	11.90%	20.47%	-9.98%
	4	30	17637.00	9550.80	5203.53	2.31	6.77%	11.09%	19.81%	-9.11%
	5	30	19522.17	11229.07	5301.31	2.63	7.23%	11.15%	18.79%	-8.95%
L6-I	1	29	6455.79	5575.83	5313.47	1.24	9.38%	9.13%	21.95%	-12.82%
	2	30	7711.38	6804.03	5279.27	1.56	11.34%	10.96%	20.80%	-10.00%
	3	30	9110.73	8155.47	5213.43	1.94	11.60%	10.67%	20.20%	-9.38%
	4	30	10418.95	9563.87	5170.07	2.31	11.13%	11.46%	19.06%	-8.84%
	5	30	12053.75	11093.03	5293.31	2.62	10.57%	9.96%	19.14%	-9.68%
avg			10334.31	6114.67	4455.79	1.74	12.68%	18.75%	38.53%	-15.75%

Table 3.10: Results of the SA minimizing logistic ratio vs total cost

same set with fewer vehicles because the routing cost grows with the fleet size, while the total quantities delivered remain relatively stable.

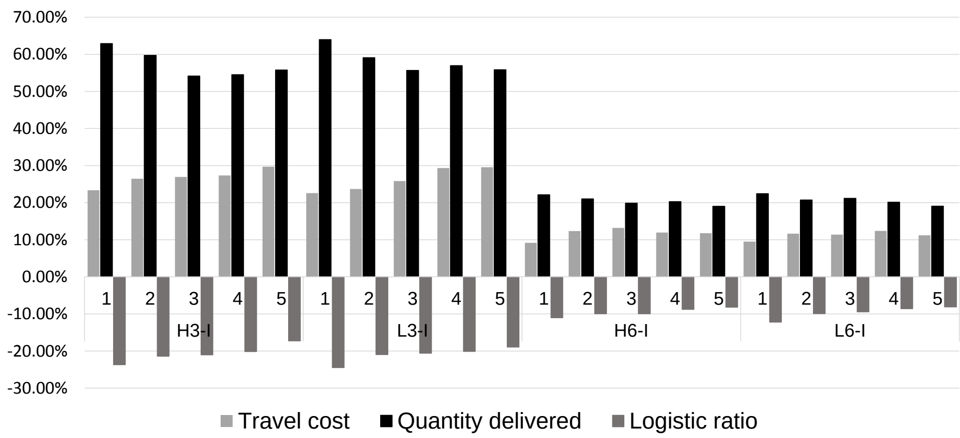


Figure 3.3: Differences of the results minimizing total cost vs logistic ratio

Finally, Table 3.11 shows the relative gaps of the best (out of five) results found by ILS and SA within 30 seconds with respect to the optimal logistic ratios reported by Archetti et al. (2017b) using a branch-price-and-cut-based method (run in a single core 3.4 GHz Intel i7-4770 processor). In that paper, the authors solved the instance set H3-I (three time periods and high inventory holding cost) and extended the instances of this set to four and five time periods, resulting in sets H4-I and H5-I, respectively. Following their developments, we also included these sets of instances in the experiment reported in Table 3.11. We grouped the instances based on the number of vehicles (nV) and the number of customers (nC), given in the first two columns. For each instance set (H3-I, H4-I and H5-I), column lr* gives the average of the optimal logistic ratios over all instances in the group. Column ILS (SA) gives the average relative difference of the logistic ratios obtained by ILS (SA) in relation to lr*. The results show that both metaheuristics are able to find optimal or near optimal solutions for most of the analyzed instances, demonstrating the ability of the developed algorithms to also address the logistic ratio as the objective function in the IRP. For the instances solved in the sets H3-I, H4-I and H5-I, ILS found solutions with average logistic ratio of 2.77, 3.39 and 3.41, respectively, while SA found solutions with average logistic ratio of 2.75, 3.35 and 3.34, for the same sets. Unfilled cells (–) are due to that Archetti et al. (2017b) did not report results for those instances. Notice that SA outperforms ILS in all instance sets. This fact can be explained to some extent by the flat structure of the objective function around the optimal solution (Archetti et al., 2017b), which suggests that the weaker changes applied in each iteration of SA helped it to adapt in a better way to the new objective function. The strength of the diversification is stronger in ILS than in SA due to the larger number of elements altered in each iteration. It is worth mentioning that the average times of the method of Archetti et al. (2017b) for sets H3-I, H4-I and H5-I are 475, 460 and 1152 seconds, respectively.

nV	nC	H3-I			H4-I			H5-I		
		lr*	**ILS**	**SA**	**lr***	**ILS**	**SA**	**lr***	**ILS**	**SA**
	5	2.54	0.79%	0.00%	2.66	1.13%	0.00%	2.59	10.42%	2.32%
1	10	1.42	0.00%	0.00%	1.43	0.70%	1.40%	1.42	0.70%	2.82%
	15	1.15	0.87%	0.00%	–	–	–	–	–	–
	5	3.18	1.57%	0.00%	3.28	0.91%	0.00%	3.19	2.19%	0.31%
2	10	1.86	2.15%	0.00%	1.87	0.53%	0.00%	1.86	1.61%	1.08%
	15	1.41	2.13%	0.00%	–	–	–	–	–	–
	5	4.19	0.00%	0.00%	4.15	2.41%	0.00%	4.26	2.11%	0.23%
3	10	2.35	0.85%	0.00%	2.27	2.20%	0.00%	2.34	3.85%	1.71%
	15	1.68	0.60%	0.00%	–	–	–	–	–	–
	5	5.06	1.38%	0.00%	5.34	1.69%	0.00%	5.31	3.01%	0.00%
4	10	2.80	0.00%	0.00%	2.78	2.16%	0.36%	2.81	2.49%	1.07%
	15	1.97	2.54%	0.51%	–	–	–	–	–	–
	5	6.02	1.00%	0.00%	6.38	1.41%	0.00%	6.10	2.13%	0.00%
5	10	3.26	1.23%	0.00%	3.27	1.22%	0.00%	3.29	1.52%	0.91%
	15	2.29	1.31%	0.00%	–	–	–	–	–	–
avg		2.75	1.09%	0.03%	3.34	1.44%	0.18%	3.32	3.01%	1.04%

Table 3.11: Comparison to the optimal logistic ratios

3.7 Final remarks

In this chapter, we presented two metaheuristic algorithms to solve the basic variant of the IRP. The algorithms are based on iterated local search and simulated annealing, respectively. A construction heuristic and a randomized variable neighborhood descent heuristic are used in both algorithms. Two different objective functions were addressed. The first is the classical minimization of the total cost while the second is the ratio between travel costs and the total quantity delivered, called logistic ratio. The results of extensive computational experiments showed that the proposed algorithms can effectively handle both objective functions and provide good feasible solutions in short running times. The results with the standard cost minimization indicate that the methods can offer different advantages according to the instance characteristic, as none of them dominated the other in the whole set of benchmark instances tested. The results minimizing the logistic ratio show that SA outperformed ILS in all sets of instances used to test the algorithms, considering the average results. All the experiments were based on 1098 problem instances from the literature, and the ILS and SA algorithms found, respectively, 289 and 283 solutions with objective values better than the best known solutions in the literature (for the first objective function). For the logistic ratio, the results show that both metaheuristics are able to find optimal or near optimal solutions for most of the analyzed instances, demonstrating the ability of the developed algorithms to also address this objective function.

Chapter 4

Formulations, branch-and-cut and a hybrid heuristic algorithm for an IRP with perishable products

In this chapter, we study an inventory routing problem in which goods are perishable. In this problem, a single supplier is responsible for delivering a perishable product to a set of customers during a given finite planning horizon. The product is assumed to have a fixed shelf-life during which it is usable and after which it must be discarded. We introduce four mathematical formulations for the problem, two with a vehicle index and two without a vehicle index, and propose branch-and-cut algorithms to solve them. In addition, we propose a hybrid heuristic based on the combination of an iterated local search metaheuristic and two mathematical programming components. We present the results of extensive computational experiments using instances from the literature as well as new larger instances. The results show the different advantages of the introduced formulations and show that the hybrid method is able to provide high-quality solutions in relatively short running times for small- and medium-sized instances while good quality solutions are found within reasonable running times for larger instances. We also adapted the proposed hybrid heuristic to solve the basic variant of the inventory routing problem. The results using standard instances show that our heuristic is also able to find good quality solutions for this problem when compared to the state-of-the-art methods from the literature.

⋆ An article based on the contents of this chapter is published as:

Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., and Morabito, R. (2020). Formulations, branch-and-cut and a hybrid heuristic algorithm for an inventory routing problem with perishable products. *European Journal of Operational Research*, 283(2):511-529.

4.1 Introduction

Research on the integration of multiple activities throughout the supply chain has increased considerably in the last decades. Today, it is well known that such integration can lead to significant advantages in both economic and performance terms. In particular, the integration of transportation and inventory management activities has been shown to provide substantial economic benefits and to improve the usage of the available resources. However, challenging problems can arise from this integration, one of which is the inventory routing problem (IRP). The IRP consists of defining the optimal replenishment plan of the customers of a supplier throughout a planning horizon as well as the routing schedule in each time period such that a given objective is optimized.

In many different industries, raw materials, as well as intermediate and final products, are often perishable. Moreover, perishability may appear in more than one activity throughout the supply chain and can influence service levels (Amorim et al., 2013). Thus, managing perishability becomes a relevant issue in the supply chain, particularly in inventory management activities. Perishability was first studied in the context of IRPs by Federgruen et al. (1986), who addressed an inventory management and distribution problem for a product that must be discarded if it is not used during a given fixed lifetime. The authors studied different patterns and policies for the distribution part of the problem. The objective was to minimize the sum of transportation and expected shortage and discarding costs.

Hemmelmayr et al. (2009) studied the problem faced by a blood bank in the distribution to hospitals. In their problem, no vehicle capacity constraints were considered (given the small size of the blood bags) but the maximum length of the routes was limited. Also, no inventory holding costs were considered since it is preferable to maintain high inventory levels rather than to experience stockouts, given the nature of the service being provided. The objective is to minimize travel costs over a finite horizon. To solve the problem, the authors proposed a basic heuristic algorithm based on a reactive visit policy, a mixed-integer programming (MIP) formulation and a variable neighborhood search approach. Le et al. (2013) studied an IRP with perishability features also motivated by a healthcare application. In their problem, it was assumed that the perishable goods have a fixed shelf-life, and they are not usable when this lifetime is exceeded. Upper bounds on the inventory levels of the customers were determined only by the perishability constraints since the discarding of products is not allowed. Thus, deliveries to customers at any given time period were limited only by the shelf-life of the goods. The objective was to minimize the sum of travel and inventory holding costs. Diabat et al. (2016) addressed the same problem as Le et al. (2013) but only minimizing travel costs.

Coelho and Laporte (2014b) considered an IRP with a fixed shelf-life perishable product with age-dependent revenues and holding costs. They presented a MIP mathematical formulation and explored different strategies to model the product consumption at the customer facilities. Mirzaei and Seifi (2015) addressed an IRP for perishable goods in which the objective function included a penalty that depends on the age of the product that is used to satisfy the demands. This penalty was included in an attempt to avoid overstocking to reduce transportation costs.

The objective was to minimize the sum of routing, inventory and penalty costs. Soysal et al. (2015) addressed an IRP with a fixed shelf-life perishable product. The authors proposed models that also considered fuel consumption and demand uncertainty. Split deliveries and backlogging of the demand were allowed as well. The objective was to minimize the sum of routing (driver wages and fuel consumption), inventory and spoilage costs. Azadeh et al. (2017) studied an IRP of a single perishable product with an exponentially decaying inventory. The authors included the possibility of transshipments between customers (performed by an outsourced third-party operator) since a single vehicle with limited capacity was considered. Backlogging was not allowed and the objective was to minimize the sum of inventory and travel costs (including transshipments costs) as well as spoilage costs. Crama et al. (2018) addressed an IRP for a single perishable product with stochastic demands (with a known probability distribution). A maximum time on the duration of the routes was imposed and no salvage value was included in their problem. Rohmer et al. (2019) addressed a two-echelon inventory routing problem for a perishable product whose age increases one unit each time period. The authors presented a MIP formulation and proposed a hybrid solution method based on the combination of an adaptive large neighborhood search metaheuristic and a mathematical model.

Shaabani and Kamalabadi (2016), Qiu et al. (2019) and Neves-Moreira et al. (2019) studied production routing problems (PRP) for perishable products. PRPs add production decisions to the IRP in an attempt to jointly optimize production, inventory and routing decisions (Adulyasak et al., 2015b; Miranda et al., 2018). Shaabani and Kamalabadi (2016) addressed the case with multiple products. In their problem, perishability was modeled as in Le et al. (2013), i.e., upper bounds on the inventory levels are determined only by the perishability constraints and discarding of products is forbidden. Qiu et al. (2019) addressed a PRP including deterioration rates and inventory holding costs that are both age-dependent. The authors tested different delivery and selling priority policies. Neves-Moreira et al. (2019) modeled the perishability feature in a PRP with multiple perishable products by setting a maximum difference between the consumption (at the customers) and production (at the plant) periods.

In this chapter, we address the IRP for a single perishable product proposed by Coelho and Laporte (2014b), which we will refer to as the PIRP (perishable IRP). In this problem, the authors model the perishability feature by considering an aging product with a fixed shelf-life as well as setting inventory holding costs and sales revenues depending on the age of the product. This type of perishability modeling is in line with the classification framework proposed by Amorim et al. (2013) for production and distribution planning. In general, perishability can be classified into three types. The first type is associated to the physical deterioration of the products as time goes by. The second type is related to the perceived value of the product for the customers, which may change or not with the product age. Finally, the third type is associated to regulations that directly influence the occurrence of the spoilage event. Thus, the PIRP can be used as a basis to model several applications involving these three types of perishability, as it only assumes that the product has a fixed lifetime and no restricting assumption is made on the age-dependent revenue and holding cost values. Examples of perishable products that deteriorate or experience a reduction in the perceived value as the expiration date approaches include milk, fresh food as well as fruits and vegetables (Abdel-Malek and Ziegler, 1988; Yu and Nagurney, 2013). Other products, such as blood products as well as some chemical and pharmaceutical products, have a relatively slow deterioration process and stable perceived value during their whole lifetime and must be discarded as soon as the expiration date is reached (Dillon et al., 2017; Chen, 2018).

The contributions of this chapter are threefold. First, we present and compare four mathematical formulations of the problem, which are solved using branch-and-cut (B&C) algorithms. We also report an inconsistency in the mathematical formulation presented by Coelho and Laporte (2014b) and show how we addressed it. Second, we develop a hybrid heuristic method based on the combination of an iterated local search (ILS) metaheuristic and two mathematical programming components. To the best of our knowledge, this is the first heuristic algorithm developed to solve the PIRP proposed by Coelho and Laporte (2014b). Also, we are not aware of any other specific solution method developed for this problem. Moreover, the models and methods presented in this chapter can be used as a basis for addressing PIRPs considering other relevant features, such as stochastic demands as in Crama et al. (2018). Third, we report the results of extensive computational experiments and introduce new large-sized problem instances. The results show the different advantages of the proposed formulations and also reveal the effectiveness of our method when solving the PIRP as well as the basic variant of the IRP. We also present a further analysis of the robustness of the algorithm's behavior.

The remaining sections of this chapter are organized as follows. In Section 4.2, we describe the problem that we address in this chapter. Section 4.3 presents the mathematical formulations introduced for the problem and the B&C algorithms used to solve them. Then, the hybrid solution method that we developed is described in detail in Section 4.4. Section 4.5 shows the computational experiments that we performed with the formulations and the hybrid method. Finally, in Section 4.6 we conclude the chapter.

4.2 Problem description

In the PIRP (Coelho and Laporte, 2014b), a supplier is responsible for delivering a single perishable product to a set of customers during a given finite multi-period planning horizon. The product is assumed to have a fixed shelf-life during which it is usable and after which it must be discarded due to its perishable nature.

The problem can be defined on a complete undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \{0, 1, \ldots, N\}$ is the vertex set and $\mathcal{E} = \{(i,j) : i, j \in \mathcal{N}, i < j\}$ is the edge set. Vertex 0 represents the supplier depot which has a homogeneous fleet of K vehicles of capacity Q, denoted by set $\mathcal{K} = \{1, \ldots, K\}$. The remaining vertices of set $\mathcal{N}$, denoted by $\mathcal{C} = \{1, \ldots, N\}$, represent the customers. Therefore, the vertex set $\mathcal{N}$ represents all the facilities of the distribution network.

The planning horizon is denoted by a set of time periods $\mathcal{T} = \{1, \ldots, T\}$. The perishable product under consideration spoils S time periods after becoming available at the supplier and its age increases by one unit in every time period. Thus, the age of the product belongs to a discrete set $\mathcal{S} = \{0, 1, \ldots, S\}$. The age of the product defines its value, according to the sales revenue u_{is} specified for each unit of age $s \in \mathcal{S}$ consumed by customer $i \in \mathcal{C}$. A travel cost c_{ij} is associated with every edge $(i, j) \in \mathcal{E}$ and an age-dependent inventory holding cost h_{is} is charged at both the supplier 0 and the customers $i \in \mathcal{C}$ for each unit of product of age $s \in \mathcal{S}$ at the end of every time period. Each customer $i \in \mathcal{C}$ has a limited storage capacity C_i and each facility $i \in \mathcal{N}$ has an initial inventory I_{i0}^0 of fresh product (of age 0) available at the beginning of the planning horizon ($t = 0$). Thus, the initial inventory will be of age 1 in the first time period of the planning horizon ($t = 1$). Each customer $i \in \mathcal{C}$ has a known demand d_i^t for the product in every time period $t \in \mathcal{T}$, which is the minimum amount of product that the supplier must guarantee to be available at the customer at that time period. In addition, the supplier produces or receives a quantity r^t of fresh products (of age 0) in each time period $t \in \mathcal{T}$. However, this quantity is available for delivery only one time period after becoming available at the supplier's facility. Table 4.1 summarizes the introduced notation.

Sets	
$\mathcal{C}$	Set of customers, indexed by i, j
$\mathcal{N}$	Set of facilities, indexed by i, j $\{0 : \text{depot}\}$
$\mathcal{E}$	Set of edges, indexed by (i, j)
$\mathcal{T}$	Set of time periods, indexed by t, m, p
$\mathcal{S}$	Set of ages of the product, indexed by s
$\mathcal{K}$	Set of vehicles, indexed by k
Parameters	
u_{is}	Revenue for product of age s at customer i
h_{is}	Inventory holding cost for product of age s at facility i
c_{ij}	Transportation cost between facilities i and j
d_i^t	Demand of customer i in time period t
r^t	Amount made available at the supplier in period t
C_i	Storage capacity of customer i
I_{i0}^0	Initial inventory at facility i
S	Maximum age of the product
Q	Capacity of the vehicles

Table 4.1: Sets and parameters of the problem

To illustrate the aging process of the inventory during the planning horizon, Figure 4.1 shows an example of the evolution of the end-of-period inventory for a given customer. Assume a maximum age of two time periods ($S = 2$), no consumptions during the planning horizon and two deliveries from the supplier of 70 and 50 units of age 1 ($s = 1$) in time periods two and three, respectively. The initial inventory of the customer consists of 100 units ($s = 0$ at $t = 0$), which become of age 1 in the first time period ($s = 1$ at $t = 1$) and then of age 2 in the second time period ($s = 2$ at $t = 2$) of the planning horizon. Notice that these 100 units of the product reached the maximum age ($s = S = 2$) in time period 2, in which they are still usable to satisfy potential demand in period 2. These units of maximum age will still be held in inventory at the end of period 2, but they will be discarded in period 3 and hence do not appear in the inventory in time period 3. Similarly, the amount received in $t = 2$, which was of age 1 ($s = 1$), becomes of age 2 ($s = 2$) in time period 3, reaching the maximum age but still being usable to satisfy potential demand in period 3. This amount will be discarded in period 4 and will not be in the usable inventory from time period 4 onwards.

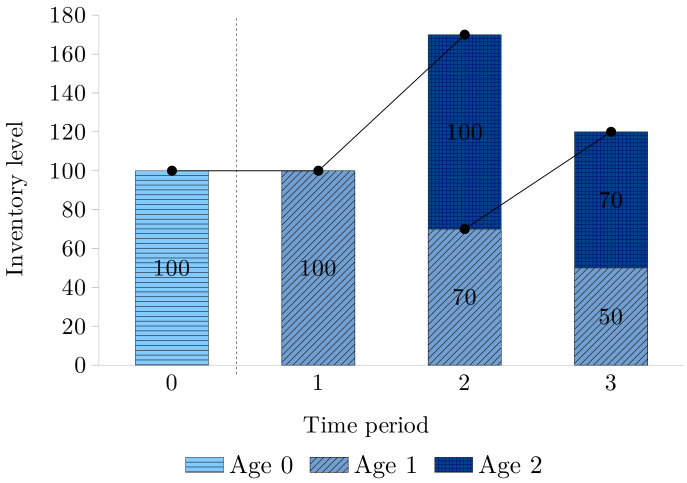


Figure 4.1: Aging process of the inventory at a customer in the PIRP

The PIRP consists of determining the time periods in which the customers will be visited; the quantity of product of each available age that will be delivered in every visit; the quantity of product of each available age that will be used to satisfy the demand; and the delivery routes to perform those visits. The objective is to maximize the total profit, given by the sales revenue minus the sum of inventory holding and routing costs. The holding costs are charged on the inventories at the end of each time period at both the supplier and customers. We consider that products of different ages share the same joint holding space at all facilities. It is also assumed that the supplier holding capacity is unbounded. In addition, according to the usual practice in the literature, we assume that the customers who receive a delivery in a given time period can use this to fulfill the demand in the same time period. As in Coelho and Laporte (2014b), we assume that the product that has reached its maximum age ($s = S$) at the end of a time period, will not go into the regular storage area, but will be kept separately in inventory to be discarded in the next period. Thus, these amounts incur the inventory holding costs but do not limit the quantity that the customer can receive in the next time period.

4.3 Mathematical formulations

This section presents the mathematical formulations we introduce for the PIRP. First, we present a corrected version of the arc-based formulation introduced by Coelho and Laporte (2014b) and show why there is an inconsistency in their formulation. Then, in the subsequent sections we present several reformulations of the problem.

4.3.1 Arc-based formulation

To formulate the PIRP using arc variables as in Coelho and Laporte (2014b), consider the following notation. First, we introduce the set $\mathcal{S}^t = \{s \in \mathcal{S} : 1 \leq s \leq t\}$, which is the subset of product ages that can be available at all facilities in time period t. This set indicates the ages that can be delivered by the supplier in each time period and also specifies the ages that can be used to satisfy the demand of the customer in the given time period. Notice that this set does not contain age 0, which is also part of the ages set $\mathcal{S}$ and is available at the supplier in each time period, given that the supplier never delivers products of age 0 to the customers because the amount made available at the supplier facility in a certain period can only be delivered in the following period. Also, let $U_i = \min\{Q, C_i\}$ be an upper bound on the amount that can be delivered to customer i in time period t. Finally, consider the following decision variables:

$x_{ij}^{kt} \in \{0,1,2\}$: number of times vehicle $k \in \mathcal{K}$ traverses edge $(i,j) \in \mathcal{E}$ in time period $t \in \mathcal{T}$;

$y_i^{kt} \in \{0,1\}$: 1 if facility $i \in \mathcal{N}$ is visited by vehicle $k \in \mathcal{K}$ in period $t \in \mathcal{T}$, 0 otherwise;

$I_{is}^t \geq 0$: inventory of age $s \in \mathcal{S}$ at facility $i \in \mathcal{N}$ at the end of time period $t \in \mathcal{T}$;

$q_{is}^{kt} \geq 0$: quantity of product of age $s \in \mathcal{S}$ delivered to customer $i \in \mathcal{C}$ by vehicle $k \in \mathcal{K}$ in period $t \in \mathcal{T}$;

$w_{is}^t \geq 0$: quantity of product of age $s \in \mathcal{S}$ used to fulfill the demand of customer $i \in \mathcal{C}$ in period $t \in \mathcal{T}$.

Given these variables, the arc-based (AB) formulation of the problem can be stated as:

$$\max \quad \sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} u_{is}w_{is}^t - \sum_{(i,j)\in\mathcal{E}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} c_{ij}x_{ij}^{kt} - \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} h_{is}I_{is}^t - \sum_{t\in\mathcal{T}} h_{00}r^t \tag{4.1}$$

$$\text{s.t.} \quad I_{0s}^t = r^t \qquad t \in \mathcal{T}, s = 0, \tag{4.2}$$

$$I_{0s}^t = I_{0,s-1}^{t-1} - \sum_{i\in\mathcal{C}}\sum_{k\in\mathcal{K}} q_{is}^{kt} \qquad t \in \mathcal{T}, s \in \mathcal{S}^t, \tag{4.3}$$

$$I_{is}^t = I_{i,s-1}^{t-1} + \sum_{k\in\mathcal{K}} q_{is}^{kt} - w_{is}^t \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t, \tag{4.4}$$

$$d_i^t = \sum_{s\in\mathcal{S}^t} w_{is}^t \qquad i \in \mathcal{C}, t \in \mathcal{T}, \tag{4.5}$$

$$\sum_{s\in\mathcal{S}^{t-1}\setminus\{S\}} I_{is}^{t-1} + \sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq C_i \qquad i \in \mathcal{C}, t \in \mathcal{T}, \tag{4.6}$$

$$\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq U_i y_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.7}$$

$$\sum_{i\in\mathcal{C}}\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq Q y_0^{kt} \qquad k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.8}$$

$$\sum_{j\in\mathcal{N}:j<i} x_{ji}^{kt} + \sum_{j\in\mathcal{N}:j>i} x_{ij}^{kt} = 2y_i^{kt} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.9}$$

$$\sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B}:j>i} x_{ij}^{kt} \leq \sum_{i\in\mathcal{B}} y_i^{kt} - y_\ell^{kt} \qquad \forall\mathcal{B} \subseteq \mathcal{C}, |\mathcal{B}| \geq 2, k \in \mathcal{K}, t \in \mathcal{T}, \ell \in \mathcal{B}, \tag{4.10}$$

$$\sum_{k\in\mathcal{K}} y_i^{kt} \leq 1 \qquad i \in \mathcal{C}, t \in \mathcal{T}, \tag{4.11}$$

$$I_{is}^t \geq 0 \qquad i \in \mathcal{N}, t \in \mathcal{T}, s \in \mathcal{S}^t, \tag{4.12}$$

$$q_{is}^{kt} \geq 0 \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}^t, \tag{4.13}$$

$$w_{is}^t \geq 0 \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t, \tag{4.14}$$

$$y_i^{kt} \in \{0,1\} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.15}$$

$$x_{ij}^{kt} \in \{0,1\} \qquad (i,j) \in \mathcal{E}: i \neq 0, k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.16}$$

$$x_{ij}^{kt} \in \{0,1,2\} \qquad (i,j) \in \mathcal{E}: i = 0, k \in \mathcal{K}, t \in \mathcal{T}. \tag{4.17}$$

The objective function (4.1) consists of maximizing the total profit, given by the total revenue minus the sum of transportation and inventory holding costs. The last term of the objective function accounts for the inventory holding cost incurred by the amount of fresh product made available at the supplier in each time period of the planning horizon. This term can be ignored as it is a constant, but for the sake of completeness, we decided to keep it in the objective function. Constraints (4.2)-(4.3) define the inventory conservation at the supplier, where the first constraint set explicitly defines the inventory of age 0 in each time period and the second constraint set defines the inventory for ages in set $\mathcal{S}^t$. Constraints (4.4) define the inventory conservation at the customers. Constraints (4.5) guarantee the fulfillment of the customer demands, which can be done with products of different ages (but only with the ages that are available in the time period of the demand). Constraints (4.6) impose that the inventory level after delivery at the customer facilities cannot exceed their storage capacity. Notice that products of different ages share the same storage space. Also, note that products of age S still available at the end of time period $t-1$ will not enter into the storage space and hence do not limit the amount that can be delivered in period t.

Constraints (4.7) permit a vehicle to perform a delivery to a specific customer only if this customer is visited by the vehicle. Constraints (4.8) guarantee that the capacity of each vehicle is respected. Constraints (4.9) ensure the flow conservation. Constraints (4.10) are subtour elimination constraints (SECs). Constraints (4.11) impose that each customer can be visited at most once in each time period. Finally, the domain of the decision variables is defined in constraints (4.12)-(4.17). Notice that when $i \neq 0$ and $j > i$, x_{ij}^{kt} can only take the values 0 or 1; if $i = 0$, then x_{ij}^{kt} can also be equal to 2, indicating that vehicle k makes a round trip between the depot and customer j in time period t. It is worth mentioning that other works in the literature have also considered delivery and inventory variables that are discretized by the different ages of the product (e.g., Rohmer et al., 2019).

This formulation has two main differences with respect to the one proposed by Coelho and Laporte (2014b). First of all, in their formulation, sums over variables of different ages (as in constraints (4.5)-(4.8) and in the objective function) consider the whole set of ages $\mathcal{S}$, instead of the subset $\mathcal{S}^t$, which we introduce here. Second, the authors define inventory conservation constraints for products of age 0 for the customers (in the form $I_{i0}^t = \sum_{k \in \mathcal{K}} q_{i0}^{kt} - d_{i0}^t, \forall i \in \mathcal{C}, t \in \mathcal{T}$), although the supplier cannot deliver these products and the customers never receive products of age 0, according to the assumptions of the problem as defined by Coelho and Laporte (2014b) via their supplier inventory constraints. These two differences can lead to an internal inconsistency in their formulation. More specifically, in the Coelho and Laporte (2014b) formulation, the variable q_{i0}^{kt} is defined, but only appears in the inventory conservation constraint for products of age 0 at the customers. Furthermore, their demand fulfillment constraints enable the satisfaction of the demand using products of age 0. As a result, the solutions of their formulation can have consumptions (w_{is}^t) and deliveries (q_{is}^t) of products of age 0 without subtracting these amounts from the supplier's inventory. This can be beneficial in a solution because the products of age 0 have a high revenue in the instances proposed by those authors. Notice that if deliveries of products of age 0 were to be allowed, the term $\sum_{i\in\mathcal{C}}\sum_{k\in\mathcal{K}} q_{i0}^{kt}$ should be subtracted from the right-hand side of constraints (4.2) and the set $\mathcal{S}^t$ should include 0.

4.3.2 Transportation formulation I

The first reformulation we propose uses decision variables that explicitly indicate the detailed use of the deliveries of each age, i.e., the time periods in which the delivery will cover all or part of the demand, as in the facility location formulation of the single item uncapacitated lot sizing problem, introduced by Krarup and Bilde (1977). For this, we introduce some additional notation. Let $T_s^t = \min\{T, t-s+S\}$ be the last time period in which a product that is of age s in time period t can be used to satisfy any demand. We also consider an additional fictitious time period $T+1$ in order to handle inventories at the end of the planning horizon. Consider the following decision variables:

$q_{is}^{ktm} \geq 0$: quantity of product of age $s \in \mathcal{S}$ delivered to customer $i \in \mathcal{C}$ by vehicle $k \in \mathcal{K}$ in period $t \in \mathcal{T}$ to cover the demand of period $m \in \{t, \dots, T_s^t + 1\}$;

$b_i^t \geq 0$: amount of the initial inventory of customer $i \in \mathcal{C}$ used to fulfill its own demand in time period $t \in \mathcal{T}$.

Note that in the definition of the variable q_{is}^{ktm}, the index age (s) refers to the age of the product at the time of the delivery. Notice that when $m = T+1$ in the delivery variables (q), it indicates that the quantity delivered will remain in the customer inventory at the end of the planning horizon. Also, when $m = (t-s+S)+1$ it means that the product will spoil and will be discarded at the customer facility in period m. Using the introduced notation and variables, the transportation formulation (TP-I) can be stated as follows:

$$\max \quad (4.1)$$

$$\text{s.t.} \quad I_{0s}^t = r^{t-s} - \sum_{i\in\mathcal{C}}\sum_{k\in\mathcal{K}}\sum_{t'=0}^{s-1}\sum_{m=t-t'}^{T_s^t+1} q_{i,s-t'}^{k,t-t',m} \qquad t \in \mathcal{T}, s \in \mathcal{S}^t, \tag{4.18}$$

$$I_{is}^t = \sum_{k\in\mathcal{K}}\sum_{t'=0}^{s-1}\sum_{m=t+1}^{T_s^t+1} q_{i,s-t'}^{k,t-t',m} \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t : s < t, \tag{4.19}$$

$$I_{is}^t = \sum_{k\in\mathcal{K}}\sum_{t'=0}^{s-1}\sum_{m=t+1}^{T_s^t+1} q_{i,s-t'}^{k,t-t',m} + I_{i0}^0 - \sum_{t'=1}^{t} b_i^{t'} \qquad i \in \mathcal{C}, t \in \mathcal{T} : t \leq S, s = t, \tag{4.20}$$

$$w_{is}^t = \sum_{k\in\mathcal{K}}\sum_{t'=0}^{s-1} q_{i,s-t'}^{k,t-t',t} \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t : s < t, \tag{4.21}$$

$$w_{is}^t = \sum_{k\in\mathcal{K}}\sum_{t'=0}^{s-1} q_{i,s-t'}^{k,t-t',t} + b_i^t \qquad i \in \mathcal{C}, t \in \mathcal{T} : t \leq S, s = t, \tag{4.22}$$

$$\sum_{s\in\mathcal{S}^{t-1}\setminus\{S\}} I_{is}^{t-1} + \sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{ktm} \leq C_i \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{4.23}$$

$$\sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{ktm} \leq U_i y_i^{kt} \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{4.24}$$

$$\sum_{i\in\mathcal{C}}\sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{ktm} \leq Q y_0^{kt} \qquad k\in\mathcal{K}, t\in\mathcal{T}, \tag{4.25}$$

$$q_{is}^{ktm} \geq 0 \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, s\in\mathcal{S}^t, m\in\{t,\ldots,T_s^t+1\}, \tag{4.26}$$

$$b_i^t \geq 0 \qquad i\in\mathcal{C}, t\in\mathcal{T}: t\leq S, \tag{4.27}$$

(4.5), (4.9)-(4.12) and (4.14)-(4.17).

Constraints (4.18) define the inventory at the supplier for all time periods and available ages (where $r^0 = I_{00}^0$). Constraints (4.19)-(4.20) define the inventory at the customers for all the different ages of the product, with (4.20) including the amount from the initial inventory that is not used to satisfy any demand. These constraints can be easily generalized for the case when the initial inventory is composed of products of different ages. Constraints (4.21)-(4.22) define the amount of product of each different age used to fulfill the demand of the customers. Notice that constraints (4.22) include the demand that can be fulfilled using the initial inventory as well. Constraints (4.23) impose that the inventory level after delivery at the customers cannot exceed their storage capacity. Constraints (4.24) allow a vehicle to perform a delivery to a specific customer only if this customer is visited by the vehicle. Constraints (4.25) guarantee that the capacity of each vehicle is respected. Finally, constraints (4.26)-(4.27) define the domain of the new decision variables. It is worth mentioning that this formulation does not correspond to the application of Krarup and Bilde's classical reformulation. Such case corresponds to formulation shown in Section 4.3.4.

4.3.3 Formulation TP-I without a vehicle index

The previous formulation can be reformulated by dropping the vehicle index of the variables, as we consider a homogeneous vehicle fleet (in both capacity and travel cost terms) and assume at most one visit to each customer in each time period. Thus, for this formulation let x_{ij}^t be an integer variable indicating the number of times a vehicle traverses edge $(i,j)\in\mathcal{E}$ in time period $t\in\mathcal{T}$, y_i^t a binary variable indicating whether or not customer $i\in\mathcal{C}$ is visited in period $t\in\mathcal{T}$, y_0^t an integer variable indicating the number of vehicles used in period $t\in\mathcal{T}$, and q_{is}^{tm} a non-negative continuous variable indicating the amount of product of age $s\in\mathcal{S}$ delivered to customer $i\in\mathcal{C}$ in time period $t\in\mathcal{T}$ to cover the demand of period $m\in\{t,\ldots,T_s^t+1\}$. The formulation, which we will refer to as TP-I-nk, can be stated as:

$$\max \quad \sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} u_{is}w_{is}^t - \sum_{(i,j)\in\mathcal{E}}\sum_{t\in\mathcal{T}} c_{ij}x_{ij}^t - \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} h_{is}I_{is}^t - \sum_{t\in\mathcal{T}} h_{00}r^t \tag{4.28}$$

$$\text{s.t.} \quad I_{0s}^t = r^{t-s} - \sum_{i\in\mathcal{C}}\sum_{t'=0}^{s-1}\sum_{m=t-t'}^{T_s^t+1} q_{i,s-t'}^{t-t',m} \qquad t\in\mathcal{T}, s\in\mathcal{S}^t, \qquad (4.29)$$

$$I_{is}^t = \sum_{t'=0}^{s-1}\sum_{m=t+1}^{T_s^t+1} q_{i,s-t'}^{t-t',m} \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t : s<t, \qquad (4.30)$$

$$I_{is}^t = \sum_{t'=0}^{s-1}\sum_{m=t+1}^{T_s^t+1} q_{i,s-t'}^{t-t',m} + I_{i0}^0 - \sum_{t'=1}^{t} b_i^{t'} \qquad i\in\mathcal{C}, t\in\mathcal{T} : t\leq S, s=t, \qquad (4.31)$$

$$w_{is}^t = \sum_{t'=0}^{s-1} q_{i,s-t'}^{t-t',t} \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t : s<t, \qquad (4.32)$$

$$w_{is}^t = \sum_{t'=0}^{s-1} q_{i,s-t'}^{t-t',t} + b_i^t \qquad i\in\mathcal{C}, t\in\mathcal{T} : t\leq S, s=t, \qquad (4.33)$$

$$\sum_{s\in\mathcal{S}^{t-1}\setminus\{S\}} I_{is}^{t-1} + \sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{tm} \leq C_i \qquad i\in\mathcal{C}, t\in\mathcal{T}, \qquad (4.34)$$

$$\sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{tm} \leq U_i y_i^t \qquad i\in\mathcal{C}, t\in\mathcal{T}, \qquad (4.35)$$

$$\sum_{j\in\mathcal{N}:j<i} x_{ji}^t + \sum_{j\in\mathcal{N}:j>i} x_{ij}^t = 2y_i^t \qquad i\in\mathcal{N}, t\in\mathcal{T}, \qquad (4.36)$$

$$Q\sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B}:j>i} x_{ij}^t \leq \sum_{i\in\mathcal{B}}(Qy_i^t - \sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{tm}) \qquad \forall\mathcal{B}\subseteq\mathcal{C}, |\mathcal{B}|\geq 2, t\in\mathcal{T}, \qquad (4.37)$$

$$y_0^t \leq K \qquad t\in\mathcal{T}, \qquad (4.38)$$

$$q_{is}^{tm} \geq 0 \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t, m\in\{t,\ldots,T_s^t+1\}, \qquad (4.39)$$

$$y_0^t \in \mathbb{Z} \qquad t\in\mathcal{T}, \qquad (4.40)$$

$$y_i^t \in \{0,1\} \qquad i\in\mathcal{C}, t\in\mathcal{T}, \qquad (4.41)$$

$$x_{ij}^t \in \{0,1\} \qquad (i,j)\in\mathcal{E}: i\neq 0, t\in\mathcal{T}, \qquad (4.42)$$

$$x_{ij}^t \in \{0,1,2\} \qquad (i,j)\in\mathcal{E}: i=0, t\in\mathcal{T}, \qquad (4.43)$$

$$(4.5), (4.12), (4.14) \text{ and } (4.27).$$

The objective function (4.28) consists of maximizing the total profit. Constraints (4.29) and (4.30)-(4.31) define the inventory level for the different ages of the product at the supplier and customers, respectively, where $r^0 = I_{00}^0$ in constraints (4.29). Constraints (4.32)-(4.33) define the amount of each different age used to fulfill the demand of the customers. Constraints (4.34) impose the maximum storage capacity at the customers. Constraints (4.35) allow to perform deliveries to a specific customer only if it is visited by a vehicle. Constraints (4.36) ensure the conservation of the flow. Constraints (4.37) are SECs and ensure that the vehicle capacities are respected as well. Constraints (4.38) limit the number of vehicles that can be used in each time period. Constraints (4.39)-(4.43) define the domain of the decision variables.

Notice that, as pointed out by Adulyasak et al. (2014a), if one divides the inequalities (4.37) by Q, they have a form similar to the generalized fractional SECs (GFSECs) for the vehicle routing problem (VRP) (Toth and Vigo, 2002). However, GFSECs in the form (4.37) are numerically more stable than the original GFSECs, which contain a fractional right-hand side.

4.3.4 Transportation formulation II

This reformulation, similar to TP-I, uses decision variables that explicitly indicate the detailed use of the deliveries of each age. However, in this case we consider implicitly the age of the product being delivered. Thus, the delivery variable is defined as follows:

$q_i^{ktpm} \geq 0$: amount of product that was made available at the supplier in period $t \in \{0\} \cup \mathcal{T}$ and was delivered to customer $i \in \mathcal{C}$ by vehicle $k \in \mathcal{K}$ in period $p \in \mathcal{T}$ to cover the demand of period $m \in \{p, \ldots, T_0^t + 1\}$.

Note that in the definition of the variable q_i^{ktpm}, the age of the product at delivery (consumption) is given by the difference between indices p (m) and t. Notice that when $t = 0$, the amount delivered comes from the initial inventory of the supplier. Notice also that, analogously to formulation TP-I, when $m = T + 1$ in the delivery variables (q) the quantity delivered will remain in the customer inventory at the end of the planning horizon and when $m = t + S + 1$ the product delivered will spoil and be discarded at the customer facility in period m. Formulations using similar facility location-based variables were presented by Neves-Moreira et al. (2019) for a PRP with perishable products and by Solyalıand Süral (2012) for the one-warehouse multi-retailer problem. The formulation, which we will refer to as TP-II, can be stated as:

$$\max \quad (4.1)$$

$$\text{s.t.} \quad I_{0s}^t = r^{t-s} - \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{K}} \sum_{p=t-s+1}^{t} \sum_{m=p}^{T_s^t+1} q_i^{k,t-s,p,m} \qquad t \in \mathcal{T}, s \in \mathcal{S}^t, \qquad (4.44)$$

$$I_{is}^t = \sum_{k \in \mathcal{K}} \sum_{p=t-s+1}^{t} \sum_{m=t+1}^{T_s^t+1} q_i^{k,t-s,p,m} \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t : s < t, \qquad (4.45)$$

$$I_{is}^t = \sum_{k \in \mathcal{K}} \sum_{p=1}^{t} \sum_{m=t+1}^{T_s^t+1} q_i^{k0pm} + I_{i0}^0 - \sum_{t'=1}^{t} b_i^{t'} \qquad i \in \mathcal{C}, t \in \mathcal{T} : t \leq S, s = t, \qquad (4.46)$$

$$w_{is}^t = \sum_{k \in \mathcal{K}} \sum_{p=t-s+1}^{t} q_i^{k,t-s,p,t} \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t : s < t, \qquad (4.47)$$

$$w_{is}^t = \sum_{k \in \mathcal{K}} \sum_{p=1}^{t} q_i^{k0pt} + b_i^t \qquad i \in \mathcal{C}, t \in \mathcal{T} : t \leq S, s = t, \qquad (4.48)$$

$$\sum_{s \in \mathcal{S}^{t-1} \setminus \{S\}} I_{is}^{t-1} + \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{k,t-s,tm} \leq C_i \qquad i \in \mathcal{C}, t \in \mathcal{T}, \qquad (4.49)$$

$$\sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{k,t-s,tm} \leq U_i y_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \qquad (4.50)$$

$$\sum_{i \in \mathcal{C}} \sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{k,t-s,tm} \leq Q y_0^{kt} \qquad k \in \mathcal{K}, t \in \mathcal{T}, \qquad (4.51)$$

$$q_i^{ktpm} \geq 0 \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \{0\} \cup \mathcal{T} \backslash \{T\},$$
$$p \in \{t+1, \dots, T_0^t\}, m \in \{p, \dots, T_0^t + 1\}, \qquad (4.52)$$

(4.5), (4.9)-(4.12), (4.14)-(4.17) and (4.27).

Constraints (4.44) calculate the inventory at the supplier for the available ages, where $r^0 = I_{00}^0$. Constraints (4.45)-(4.46) define the inventory at the customers for all the different ages of the product. Constraints (4.47)-(4.48) state that the demand of the customers can be satisfied using products of all the available ages. Notice that constraints (4.48) include the demand that can be fulfilled using the initial inventory as well. Constraints (4.49) impose the maximum storage capacity after delivery at the customer facilities. Constraints (4.50) allow deliveries to a customer by a specific vehicle only if it is visited by the same vehicle. Constraints (4.51) guarantee that the capacity of each vehicle is respected. Finally, constraints (4.52) define the domain of the new decision variable.

4.3.5 Formulation TP-II without a vehicle index

As in Section 4.3.3, an additional formulation can be obtained by dropping the vehicle index of the variables for cases in which the vehicle fleet is considered to be homogeneous and at most a single visit is allowed to each customer in each time period. For this formulation, let q_i^{tpm} be a non-negative continuous variable indicating the amount of product that was made available at the supplier in time period $t \in \{0\} \cup \mathcal{T}$ and that was delivered to customer $i \in \mathcal{C}$ in period $p \in \mathcal{T}$ to cover the demand of period $m \in \{p, \dots, T_0^t + 1\}$. Then, using this variable the formulation (TP-II-nk) can be stated as:

$$\max \quad (4.28)$$

$$\text{s.t.} \quad I_{0s}^t = r^{t-s} - \sum_{i \in \mathcal{C}} \sum_{p=t-s+1}^{t} \sum_{m=p}^{T_s^t+1} q_i^{t-s,p,m} \qquad t \in \mathcal{T}, s \in \mathcal{S}^t, \qquad (4.53)$$

$$I_{is}^t = \sum_{p=t-s+1}^{t} \sum_{m=t+1}^{T_s^t+1} q_i^{t-s,p,m} \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t : s < t, \qquad (4.54)$$

$$I_{is}^t = \sum_{p=1}^{t} \sum_{m=t+1}^{T_s^t+1} q_i^{0pm} + I_{i0}^0 - \sum_{t'=1}^{t} b_i^{t'} \qquad i \in \mathcal{C}, t \in \mathcal{T} : t \leq S, s = t, \qquad (4.55)$$

$$w_{is}^t = \sum_{p=t-s+1}^{t} q_i^{t-s,p,t} \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t : s < t, \qquad (4.56)$$

$$w_{is}^t = \sum_{p=1}^{t} q_i^{0pt} + b_i^t \qquad i \in \mathcal{C}, t \in \mathcal{T} : t \leq S, s = t, \qquad (4.57)$$

$$\sum_{s \in \mathcal{S}^{t-1} \backslash \{S\}} I_{is}^{t-1} + \sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{t-s,tm} \leq C_i \qquad i \in \mathcal{C}, t \in \mathcal{T}, \qquad (4.58)$$

$$\sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_i^{t-s,tm} \leq U_i y_i^t \qquad i\in\mathcal{C}, t\in\mathcal{T}, \quad (4.59)$$

$$Q\sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B}:j>i} x_{ij}^t \leq \sum_{i\in\mathcal{B}}(Qy_i^t - \sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_i^{t-s,tm}) \qquad \forall\mathcal{B}\subseteq\mathcal{C}, |\mathcal{B}|\geq 2, t\in\mathcal{T}, \quad (4.60)$$

$$q_i^{tpm} \geq 0 \qquad i\in\mathcal{C}, t\in\{0\}\cup\mathcal{T}\backslash\{T\}, p\in\{t+1,\ldots,T_0^t\}, m\in\{p,\ldots,T_0^t+1\}, \quad (4.61)$$

$$(4.5), (4.12), (4.14), (4.27), (4.36), (4.38) \text{ and } (4.40)\text{-}(4.43).$$

Constraints (4.53) and (4.54)-(4.55) define the inventory levels of the available ages of the product at the supplier and customers, respectively, where $r^0 = I_{00}^0$ in constraints (4.53). Constraints (4.56)-(4.57) specify that the demand of the customers can be satisfied using products of all the available ages. Constraints (4.58) enforce the maximum storage capacity at the customer facilities. Constraints (4.59) allow deliveries to customers in a given time period if they are visited by a vehicle in the same time period. Constraints (4.60) are the GFSECs and guarantee that the capacity of each vehicle is respected as well. Finally, constraints (4.61) define the domain of the new decision variable.

4.3.6 Branch-and-cut algorithms

Given that all the presented formulations contain an exponentially large number of SECs, we must apply a B&C algorithm to solve them. These constraints are dropped from the formulations and added in an iterative fashion every time they are violated at the nodes of the branch-and-bound (B&B) tree. In this section we provide the details of our B&C approaches for both the formulations with and without a vehicle index as well as further improvements.

4.3.6.1 Branch-and-cut for the vehicle index formulations

To solve the formulations with a vehicle index, we use an exact separation algorithm that solves a series of minimum $s-t$ cut problems to detect violated SECs for each vehicle in each time period of the planning horizon. At a given node of the B&B tree, let $\bar{y}_i^{kt}$ and $\bar{x}_{ij}^{kt}$ denote the values of the visit (y) and flow variables (x) of the solution, respectively. A graph for vehicle k in time period t is constructed from the set of nodes where $\bar{y}_i^{kt} > 0$, setting the weights of the graph edges to $\bar{x}_{ij}^{kt}$, $\forall(i,j)\in\mathcal{E}$. Then, for each customer node of the constructed graph, we solve a minimum $s-t$ cut problem, setting the supplier node as the source node (s) and the customer node as the sink node (t). A violated SEC is identified if the capacity of the minimum cut is less than $2\bar{y}_i^{kt}$ (Adulyasak et al., 2014a). If a subtour on a set of nodes $\mathcal{B}\subseteq\mathcal{C}$ is found for vehicle k in period t, we add constraints (4.10) with $\ell = \arg\max_{i\in\mathcal{B}}\{\bar{y}_i^{kt}\}$ to the formulation, for all vehicles and time periods of the planning horizon. To solve the minimum $s-t$ cut problem, we used the Concorde solver (Applegate et al., 2018).

These SECs are separated only at the root node and then every time an integer solution is found at a node of the B&B tree, to avoid generating too many cuts in the tree. Notice that constraints (4.10) can be added to the formulation in many different ways, among which we tested: adding the cut only for the specific vehicle and time period for which it was violated;

adding the cut for all vehicles in the same time period in which the violated cut was identified; and, finally, adding the cut for all vehicles and time periods. The latter strategy resulted in a slightly better performance. Regarding the selection of the customer $\ell \in \mathcal{B}$ for which the cut would be set, we tried including the cut only for the customer ℓ such that $\ell = \arg\max_{i \in \mathcal{B}}\{\bar{y}_i^{kt}\}$ and, for every customer in the identified subset of customers $\mathcal{B}$. In this case, the former strategy resulted in a better performance of the B&C algorithms.

4.3.6.2 Branch-and-cut for the formulations without a vehicle index

To separate GFSECs (4.37) and (4.60) of formulations TP-I-nk and TP-II-nk, respectively, we use the separation package developed by Lysgaard et al. (2004) for the VRP, as in Adulyasak et al. (2014a). This algorithm consists of four heuristic algorithms, which are applied sequentially. One of these heuristics is an exact separation algorithm when all the flow variables (x) take integer values.

For a given solution, we call the algorithm for each time period $t \in \mathcal{T}$. At a given node of the B&B tree, let $\bar{y}_i^{kt}$, $\bar{x}_{ij}^{kt}$ and $\bar{q}_{is}^{tm}$ ($\bar{q}_i^{tpm}$) denote the values of variables y_i^{kt}, x_{ij}^{kt} and q_{is}^{tm} (q_i^{tpm}) for the formulation TP-I-nk (TP-II-nk). The input required by the separation package (a VRP solution) in period t is constructed considering customer nodes with $\bar{y}_i^{kt} > 0$, setting the weight of each edge (i,j) to $\bar{x}_{ij}^{kt}$ and setting the delivery quantity for customer i to $\sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_{is}^{tm}$ and to $\sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{t-s,tm}$ for the formulations TP-I-nk and TP-II-nk, respectively.

Similar to the formulations with a vehicle index, we separate GFSECs only at the root node and then whenever an integer solution is found at a given node of the B&B tree to avoid generating too many cuts. Every time we identify a violated cut, we add the corresponding constraint for all time periods. In addition, to further strengthen formulations TP-I-nk and TP-II-nk, we included the following SECs, as used in the formulations with a vehicle index (AB, TP-I, TP-II):

$$\sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}: j > i} x_{ij}^t \leq \sum_{i \in \mathcal{B}} y_i^k - y_\ell^k \qquad \forall \mathcal{B} \subseteq \mathcal{C}, |\mathcal{B}| \geq 2, k \in \mathcal{K}, \ell \in \mathcal{B}. \tag{4.62}$$

Using these together with GFSECs resulted in an improved performance of the formulations without a vehicle index. These last SECs are separated as described in Section 4.3.6.1.

4.3.7 Valid inequalities

We can further strengthen the formulations (AB, TP-I, TP-II, TP-I-nk and TP-II-nk) by including some valid inequalities. All these inequalities have been used in previous works (Archetti et al., 2007; Engineer et al., 2012; Coelho and Laporte, 2014a; Desaulniers et al., 2016), and can also be used in our formulations. Constraints (4.63) and (4.64) enforce the relation between the routing variables (x) and the visit variables (y):

$$x_{0i}^{kt} \leq 2y_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.63}$$

$$x_{ij}^{kt} \leq y_i^{kt} \qquad i,j \in \mathcal{E}: i \neq 0, k \in \mathcal{K}, t \in \mathcal{T}. \tag{4.64}$$

The respective counterparts of these inequalities for the formulations without a vehicle index are obtained by dropping the vehicle index from the variables.

Symmetry breaking constraints can be included in the vehicle index formulations in the presence of identical vehicles, as follows:

$$y_i^{kt} \leq \sum_{j \in \mathcal{C}: j<i} y_j^{k-1,t} \qquad i \in \mathcal{C}\backslash\{1\}, k \in \mathcal{K}\backslash\{1\}, t \in \mathcal{T}, \tag{4.65}$$

$$y_0^{kt} \leq y_0^{k-1,t} \qquad k \in \mathcal{K}\backslash\{1\}, t \in \mathcal{T}. \tag{4.66}$$

We can use three additional sets of valid inequalities. The first set corresponds to inequalities on the minimum number of visits to a single customer up to a given time period. Let $\bar{I}_i^t = \max\left\{0, I_{i0}^0 - \sum_{p=1}^t d_i^p\right\}$ be the minimum amount from the initial inventory that must remain at customer i at the end of time period t, with $\bar{I}_i^t = 0, \forall i \in \mathcal{C}, t \in \mathcal{T}, t > S$ and $\bar{I}_i^0 = I_{i0}^0$. Let $\bar{d}_i^t$ be the minimum residual demand of customer i in time period t, where $\bar{d}_i^t = \max\left\{0, d_i^t - \bar{I}_i^{t-1}\right\}$, $\forall i \in \mathcal{C}, t \in \mathcal{T}, t \leq S$ and $\bar{d}_i^t = d_i^t, \forall i \in \mathcal{C}, t \in \mathcal{T}, t > S$. Then, considering all the residual demands of a customer i up to a given time period t and the maximum size of a single delivery to the customer, it is possible to compute a lower bound on the number of visits to the customer up to that time period. This lower bound is given by $LB1_i^t = \lceil \sum_{p=1}^t \bar{d}_i^p / U_i \rceil$. Now, it follows that the following inequalities are valid:

$$\sum_{p=1}^{t} \sum_{k \in \mathcal{K}} y_i^{kp} \geq LB1_i^t, \qquad i \in \mathcal{C}, t \in \mathcal{T}. \tag{4.67}$$

For the formulations without a vehicle index, the left-hand side of the inequalities is replaced by the term $\sum_{p=1}^t y_i^p$.

The second set corresponds to inequalities on the minimum number of routes up to a given time period. These can be obtained by summing over the residual demands of all the customers up to a time period. Thus, a lower bound on the minimum number of routes to serve the residual demands of all customers up to time period t is given by $LB2^t = \lceil \sum_{i \in \mathcal{C}} \sum_{p=1}^t \bar{d}_i^p / Q \rceil$. Then, the following inequalities are valid:

$$\sum_{p=1}^{t} \sum_{k \in \mathcal{K}} y_0^{kp} \geq LB2^t, \qquad t \in \mathcal{T}. \tag{4.68}$$

Analogously, for the formulations without a vehicle index, the left-hand side of the inequalities becomes $\sum_{p=1}^t y_0^p$.

The final set generalizes inequalities (4.67) by considering any time interval $[t_1, t_2]$, $\forall t_1, t_2 \in \mathcal{T}, 1 < t_1 < t_2 \leq T$. Let $\bar{d}_i^{t_1 t_2} = \sum_{t=t_1}^{t_2} d_i^t$ denote the sum of demands over time periods t_1 to t_2 and $I_i^t = \sum_{s \in \mathcal{S}^t} I_{is}^t$ denote the sum of inventory variables for customer $i \in \mathcal{C}$ in time period $t \in \mathcal{T}$. Then

$$\left\lceil \frac{\bar{d}_i^{t_1 t_2} - I_i^{t_1 - 1}}{U_i} \right\rceil$$

is a lower bound on the number of visits to customer $i \in \mathcal{C}$ from t_1 to t_2. Notice that for $t_1 = 1$

the resulting inequalities correspond to (4.67). For $t_1 > 1$ it results in a nonlinear bound given the presence of the inventory variables. However, as shown by Engineer et al. (2012), linear inequalities can be derived by appropriately bounding $I_i^{t_1-1}$. Assume that $I_i^{t_1-1} = C_i - d_i^{t_1-1}$, i.e., the inventory at the end of period $t_1 - 1$ is at its maximum level. Then, given that $I_i^{t_1-1} \leq C_i - d_i^{t_1-1}$,

$$\sum_{t=t_1}^{t_2} \sum_{k \in \mathcal{K}} y_i^{kt} \geq \left\lceil \frac{\bar{d}_i^{t_1 t_2} - (C_i - d_i^{t_1-1})}{U_i} \right\rceil \qquad i \in \mathcal{C}, 1 < t_1 < t_2 \leq T \tag{4.69}$$

is a valid inequality. Similarly, for the formulations without a vehicle index, the left-hand side of (4.69) is changed to $\sum_{t=t_1}^{t_2} y_i^t$.

4.4 Optimization-based iterated local search

In this section, we present the hybrid algorithm that we propose to solve the PIRP. This algorithm is based on the ILS metaheuristic for the basic variant of the IRP presented in Chapter 3. The basic idea of ILS is to iteratively apply a local search algorithm to solutions resulting from the perturbation of the previously visited local optima, which leads to a randomized search in the space of local optimal solutions (Lourenço et al., 2003). The ILS metaheuristic presented in Chapter 3 uses a multi-start randomized variable neighborhood descent (RVND) as the local search component and a multi-operator algorithm as perturbation mechanism. However, since the PIRP and the basic variant of the IRP have some fundamental differences (e.g., age-specific delivery, consumption and inventory decisions in the PIRP), several non-trivial adjustments had to be performed. First of all, in the proposed method the various decisions of the problem are handled by different components. On the one hand, routing decisions (x) are managed by the local search phase of the method while the visit variables (y) are mostly handled in the perturbation phase. On the other hand, a multi-commodity flow (MCF) problem formulation is used to determine the optimal values of the continuous variables (q, w and I) for a given set of visit variables (y). Additionally, a MIP formulation that can remove and insert customers from a solution given as input is used as a solution improvement (SI) step in the final phase of the method. An overview of the proposed method is shown in Algorithm 4.1.

The algorithm starts with an initial feasible solution (line 2), which is generated using the construction heuristic that will be described in Section 4.4.1. If the construction heuristic cannot find a feasible solution, the algorithm stops; otherwise, the search process continues. A RVND heuristic is used as local search algorithm (lines 4 and 7), and a multi-operator algorithm is used as a perturbation mechanism (line 6). The continuous variables (deliveries, consumptions and inventories) of the solution are then optimized by solving a MCF formulation (line 8). The acceptance criterion admits the resulting solution only if it is better than the current best solution (line 9). Finally, after reaching a stopping criterion, the method applies the SI formulation (line 11). All components of the hybrid method are described in detail in the following sections.

In the description of the hybrid method, we use the subsequent notation. Given a solution O, we denote by $\bar{I}_{is}^t$, $\bar{q}_{is}^{kt}$, $\bar{w}_{is}^t$ and $\bar{y}_i^{kt}$ the values of its inventory, delivery, consumption and visit variables, respectively. In addition,

Algorithm 4.1: Optimization-based iterated local search

begin
  $O^0 \leftarrow$ `construction_heuristic()`;
  if $O^0 \neq \emptyset$ **then**
    $O^* \leftarrow$ `rvnd_heuristic(`O^0`)`;
    while *stop criterion is not met* **do**
      $O^{'} \leftarrow$ `perturbation(`O^*`)`;
      $O^{'} \leftarrow$ `rvnd_heuristic(`$O^{'}$`)`;
      $O^{'} \leftarrow$ `optimize_amounts(`$O^{'}$`)`;
      if $f(O^{'}) > f(O^*)$ **then** $O^* \leftarrow O^{'}$;
    end
    $O^* \leftarrow$ `SI_formulation(`O^*`)`;
  end
end

- $\mathcal{R}(O)$ is the set of all vehicle routes of the solution;
- $\mathcal{C}^t(O) = \{i \in \mathcal{C} \colon \sum_{k \in \mathcal{K}} \bar{y}_i^{kt} = 1\}$ is the set of customers visited by routes of the solution in time period t;
- $\mathcal{T}_i(O) = \{t \in \mathcal{T} \colon \sum_{k \in \mathcal{K}} \bar{y}_i^{kt} = 1\}$ is the set of time periods in which customer i is visited by routes of the solution.

Also, given a route $r \in \mathcal{R}(O)$ of the solution,

- $t(r)$ is the time period of the route; and
- $\mathcal{C}(r)$ is the set of customers visited by the route.

4.4.1 A construction heuristic for the PIRP

To obtain feasible solutions, we devised a decomposition construction heuristic which iteratively separates the decisions of the problem into two phases. In the first phase, the heuristic defines the size of the potential delivery to each customer and assigns a priority to each one of them. Then, in the second phase, feasible delivery routes are designed to deliver the amounts set in the first phase.

The heuristic starts by using the initial inventory of each customer to satisfy the maximum number of demands. First, let H_i^t be the usable amount remaining from the initial inventory at customer $i \in \mathcal{C}$ at the end of period $t \in \mathcal{T}$:

$$H_i^t = \max\left\{0, I_{i0}^0 - \sum_{p=1}^{t} d_i^p\right\},$$

with $H_i^0 = I_{i0}^0$ and $H_i^t = 0$ for $t \geq S$.

The values for the consumption variables (w_{is}^t) are set as follows:

$$\bar{w}_{is}^t = \begin{cases} \min\{H_i^{t-1}, d_i^t\} & \text{if } t \leq S, s = t, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i \in \mathcal{C}, t \in \mathcal{T}, s \leq S^t. \tag{4.70}$$

Then, the heuristic computes the aggregated inventory levels I_i^t for each customer $i \in \mathcal{C}$ at the end of each time period $t \in \mathcal{T}$ given the initial consumptions, as follows:

$$I_i^t = \begin{cases} I_{i0}^0 - \sum_{p=1}^t \bar{w}_{ip}^p & \text{if } t < S, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i \in \mathcal{C}, t \in \mathcal{T}. \tag{4.71}$$

These inventory levels will be updated at the end of each iteration based on the deliveries and consumptions. The aggregated inventory levels are used to determine the delivery sizes in each time period given that the amount a customer can receive is bounded by the holding capacity and the aggregated inventory level at the end of the previous time period. Notice that, initially, $I_i^t = 0$ for $t = S$ since the initial inventory will spoil at the end of this time period and, as stated in Section 4.2, the spoiled inventory does not limit the amount that the customer can receive in the next period.

Using these values, the heuristic performs one iteration for each time period $t \in \mathcal{T}$, starting from $t = 1$. In the first phase of iteration t, the heuristic sets a potential delivery quantity to each customer by computing the difference between its capacity and the aggregated inventory level in the previous time period, also respecting the vehicle capacity. To simplify the heuristic, we apply a greedy approach in which all the deliveries are of the freshest possible product, which in our case is product of age $s = 1$. Therefore, the potential delivery ($\tilde{q}$) to each customer i is set as:

$$\tilde{q}_{i1}^t = \min\{\texttt{ratio\_demand} \times (C_i - I_i^{t-1}), Q\}, \tag{4.72}$$

where `ratio_demand` $\in (0, 1]$ is a parameter that defines the proportion of the maximum possible quantity that will be actually delivered. Next, the priority π_i of customer i is set as the number of upcoming `look_ahead` periods (including t) in which its demand is not fully covered yet, i.e., π_i is the number of times in which $\sum_{s \in \mathcal{S}^p} \bar{w}_{is}^p < d_i^p$, for $p = t, \ldots, \min\{T, t + \texttt{look\_ahead}\}$, $\forall i \in \mathcal{C}$. The value of `look_ahead` determines how much to look forward in the planning horizon, trying to anticipate forthcoming stockouts.

After defining these deliveries and priorities, the second phase of iteration t starts. It consists of determining one or more vehicle routes using a nearest-neighbor insertion heuristic that first routes customers with higher priority as long as the insertion satisfies the vehicle capacity. At most K routes can be defined in this phase. Then, given the deliveries actually performed, we set the values of $\bar{q}_{i1}^{kt}$ and update the values of the consumption (using the first-in first-consumed (FIFC) rule) and inventory variables. Finally, a new iteration is started for the next period $(t + 1)$, until reaching time period T.

A pseudo-code of the heuristic is given in Algorithm 4.2. Since the heuristic runs in a short time, it was defined inside two outer loops, exploring different values for `ratio_demand` and `look_ahead`, with the aim of finding a reasonably good feasible solution, as in Chapter 3. Furthermore, at the end of the execution of the heuristic, the continuous variables of the best feasible solution found (if any) are optimized using the MCF formulation of Section 4.4.4. Finally, as will be shown in Section 4.5, this heuristic was able to find feasible solutions for all the benchmark instances used in this chapter.

Algorithm 4.2: Construction heuristic for the PIRP

```
begin
    O^* ← ∅;
    Use initial inventory to set as many consumptions (\bar{w}) as possible, using (4.70);
    Compute aggregated inventory levels I_i^t as in (4.71), for all i ∈ \mathcal{C} and t ∈ \mathcal{T};
    ratio_demand ← 1.0;
    while ratio_demand > 0 do
        look_ahead ← 0;
        while look_ahead ≤ S do
            for t ∈ \mathcal{T} do
                for i ∈ \mathcal{C} do
                    \tilde{q}_{i1}^t ← min{ratio_demand × (C_i − I_i^{t−1}), Q};
                    π_i ← 0;
                    for p = t, ..., min{T, t + look_ahead} do
                        if \sum_{s∈\mathcal{S}^p} \bar{w}_{is}^p < d_i^p then π_i ← π_i + 1 ;
                    end
                end
                Apply a nearest-neighbor insertion heuristic, routing customers with higher π_i
                  first;
                For all routed customers, set the corresponding \bar{q}_{i1}^{kt} values (equal to \tilde{q}_{i1}^{kt}) and
                  compute the corresponding \bar{w}_{is}^t values using the FIFC rule and update \bar{I}_{is}^t and I_i^t;
            end
            Update best feasible solution O^*;
            look_ahead ← look_ahead + 1;
        end
        ratio_demand ← ratio_demand − 0.1;
    end
    if O^* ≠ ∅ then O^* ← optimize_amounts(O^*) ;
end
```

4.4.2 Randomized variable neighborhood descent heuristic

For the local search procedure of the proposed method, we use a variable neighborhood descent heuristic (Mladenović and Hansen, 1997) with random neighborhood ordering as in Chapter 3. In this algorithm, local search operators are selected randomly from a predefined set and applied to the incumbent solution until none of them can improve it.

In our method, we use the local search phase to handle the routing decisions of the solution. For this, the RVND heuristic uses the following classical VRP operators: Or-opt-k, $k \in \{1, 2, 3\}$; Shift(k), $k \in \{1, 2, 3\}$; Swap(k_1, k_2), $k_1, k_2 \in \{1, 2\}, k_1 \geq k_2$; and k-opt, $k \in \{2, 3\}$. All operators explore the search space using the first improvement strategy, allowing only feasible solutions in the search process.

4.4.3 Perturbation mechanism

Since in the PIRP there are different decisions that must be made simultaneously, we designed a perturbation algorithm that can change multiple attributes of a solution in a single call. The algorithm uses the following operators to modify the visit and delivery decisions of an input solution O.

1. Insert visits: choose randomly a route $r \in \mathcal{R}(O)$ and customer i such that $i \notin \mathcal{C}^{t(r)}(O)$. The customer is inserted into the cheapest insertion position in the route. Then the values of $\bar{q}$, $\bar{w}$ and $\bar{I}$ are re-optimized using the MCF formulation;

2. Remove visits: choose a random route $r \in \mathcal{R}(O)$ and a customer $i \in \mathcal{C}(r)$ and then remove i from r. After that, the values of $\bar{q}$, $\bar{w}$ and $\bar{I}$ are re-optimized using the MCF formulation;

3. Move visit: choose a random route $r \in \mathcal{R}(O)$ and a customer $i \in \mathcal{C}(r)$ such that $|\mathcal{T}_i(O)| < T$, i.e., a customer that is not visited in every time period of the planning horizon. Then, the visit to i is removed from r and inserted into the cheapest position of a route of a period $p \in \mathcal{T} \backslash \mathcal{T}_i(O)$, choosing both, p and the route, at random. Finally, the values of $\bar{q}$, $\bar{w}$ and $\bar{I}$ are re-optimized using the MCF formulation;

4. Reduce deliveries: choose a random route $r \in \mathcal{R}(O)$ and a delivery (of a certain age s) to a customer $i \in \mathcal{C}(r)$ such that the amount delivered is not completely consumed by the customer. This can happen, for instance, when it is profitable to accumulate inventory at the customer to save holding costs at the supplier. Then, the delivery is reduced by the amount not consumed by the customer. Both the customer and the delivery to be reduced are chosen at random.

After applying each operator, the objective function value of the solution is recomputed. The aim of these operators is twofold. First, helping to determine the periods in which each customer must be visited and, second, creating slack in the routes for the local search heuristic. In the Remove and Move operators, infeasible solutions are rejected. In such a case, the operator chooses another customer of the same route. If all customers of the chosen route are unsuccessfully explored (resulting in infeasible solutions), the operator chooses another route and the process is applied in the same fashion.

Note that the performance of an ILS-based algorithm is strongly related to the strength of its perturbation mechanism given that it defines much of the behavior of the method. This mechanism must be able to diversify the search process without turning it into a randomized restart search. For this purpose, we use the parameter `max_perturb`, which defines the maximum number of elements of the solution that can be changed each time the perturbation mechanism is called. Thus, similar to the RVND heuristic, our perturbation algorithm can use multiple operators in a single call, applying one operator at a time (changing at most one element of the solution) until either the number of changes performed to the solution reaches `max_perturb` or none of the operators can change the solution.

4.4.4 Multi-commodity flow (MCF) formulation

Given the values of $\bar{y}$ from a solution O, one can determine the optimal values for the delivery, consumption and inventory variables that maximize the total profit by solving the following MCF problem formulation:

$$\max \quad \sum_{i \in \mathcal{C}} \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}^t} u_{is} w_{is}^t - \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}^t} h_{is} I_{is}^t \tag{4.73}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}^t} q_{is}^{kt} \leq U_i \bar{y}_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.74}$$

$$\sum_{i \in \mathcal{C}} \sum_{s \in \mathcal{S}^t} q_{is}^{kt} \leq Q \bar{y}_0^{kt} \qquad k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.75}$$

(4.2)-(4.6) and (4.12)-(4.14),

where $\bar{y}_0^{kt} = 1$ indicates that vehicle k is used in time period t and $\bar{y}_i^{kt} = 1$ indicates that vehicle k visits customer i in time period t. The objective function (4.73) consists of maximizing the total profit, given by the total revenue minus the total inventory cost. Constraints (4.74) allow a vehicle to perform a delivery to a specific customer in a given time period only if the customer is visited by the vehicle in that time period in the solution O. Finally, constraints (4.75) impose the vehicle capacity. This linear program (LP) that can be solved using a general-purpose solver.

Notice that empty visits can result from this phase, i.e., cases with $\sum_{s \in \mathcal{S}^t} q_{is}^{kt} = 0$ and $\bar{y}_i^{kt} = 1$ for a given $i \in \mathcal{C}$, $k \in \mathcal{K}$, $t \in \mathcal{T}$. In such a case, the customer is removed from the route and the objective function value of the solution is updated.

4.4.5 Solution improvement (SI) formulation

As an improvement step, we use a MIP formulation for a customer assignment problem, as in Archetti et al. (2012). This model can be used to remove and insert customers into a given solution O. Let Δ_i^{kt} be the savings in the travel cost when customer i is removed from the route of vehicle k in time period t. This value is computed as $c_{hi} + c_{ij} - c_{hj}$, where h and j are, respectively, the predecessor and successor of the customer in the route. We set Δ_i^{kt} as 0 when the customer is not visited by vehicle k in time period t (i.e., $\Delta_i^{kt} = 0$ if $\bar{y}_i^{kt} = 0$). Similarly, let Γ_i^{kt} be the cost of inserting customer i into its cheapest position in the route of vehicle k in time period t. Γ_i^{kt} equals 0 for those customers that are already visited by vehicle k in time period t. The formulation uses two binary decision variables. Let δ_i^{kt} be a binary variable equal to 1 if and only if customer i is removed from the route of vehicle k in period t, and let γ_i^{kt} be a binary variable equal to 1 if and only if customer i is inserted into the route of vehicle k in time period t. Then, the solution improvement formulation can be stated as follows:

$$\max \quad \sum_{i \in \mathcal{C}} \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}^t} u_{is} w_{is}^t - \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}^t} h_{is} I_{is}^t + \sum_{i \in \mathcal{C}} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \Delta_i^{kt} \delta_i^{kt} - \sum_{i \in \mathcal{C}} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \Gamma_i^{kt} \gamma_i^{kt} \tag{4.76}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_{is}^{ktm} \leq U_i (\bar{y}_i^{kt} - \delta_i^{kt} + \gamma_i^{kt}) \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.77}$$

$$\sum_{i \in \mathcal{C}} \sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_{is}^{ktm} \leq Q \bar{y}_0^{kt} \qquad k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.78}$$

$$\gamma_i^{kt} \leq 1 - \bar{y}_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.79}$$

$$\delta_i^{kt} \leq \bar{y}_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.80}$$

$$\gamma_i^{kt} \leq \bar{y}_0^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{4.81}$$

$$\sum_{i \in \mathcal{C}} (\delta_i^{kt} + \gamma_i^{kt}) \leq \beta \qquad k \in \mathcal{K}, t \in \mathcal{T}, \qquad (4.82)$$

$$\sum_{k \in \mathcal{K}} (\bar{y}_i^{kt} - \delta_i^{kt} + \gamma_i^{kt}) \leq 1 \qquad i \in \mathcal{C}, t \in \mathcal{T}, \qquad (4.83)$$

$$\delta_i^{kt} \in \{0, 1\} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \qquad (4.84)$$

$$\gamma_i^{kt} \in \{0, 1\} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \qquad (4.85)$$

(4.5), (4.12), (4.14), (4.18)-(4.23) and (4.26)-(4.27).

The objective function (4.76) consists of maximizing the total profit, given by the total revenue minus the total inventory cost plus the difference between the savings and additional travel cost given by the removal and insertion operations, respectively. Constraints (4.77) allow vehicle k to perform deliveries to customer i in period t only if either this customer is already visited by the vehicle in the solution and it was not removed from the route or if the customer was inserted into the route of the vehicle in the given time period. Constraints (4.78) impose the vehicle capacity. Constraints (4.79) ensure that if a customer is already visited by a route, it cannot be reinserted into the same route. Analogously, constraints (4.80) allow the removal of a customer from a route only if it is visited by the route. Constraints (4.81) forbid the insertion of customers into routes of vehicles that are not used in the given time period. Notice that if more than one visit is removed or inserted from a vehicle route, then the values of Δ and Γ provide only an approximation of the actual routing costs. For this reason, we impose constraints (4.82) which limit the number of changes that can be performed to every single route to a value β. Constraints (4.83) impose that each customer can be visited at most once in each time period. Finally, constraints (4.84) and (4.85) define the domain of the removal and insertion decision variables.

Note that to model the inventory part of this model, we use delivery variables as in the formulation TP-I (Section 4.3.2) given that we need delivery variables with a vehicle index and, as will be shown in the computational experiments, formulation TP-I had a slightly better performance than TP-II. It is worth mentioning that several different solution improvement phases using mathematical programming components have been used within hybrid heuristic methods in the literature, see e.g., Archetti et al. (2012), Larrain et al. (2017), Archetti et al. (2017a) and Neves-Moreira et al. (2019).

4.4.6 Details on the computational implementation

In this section we provide some details on the computational implementation of the optimization-based ILS. Although the presented hybrid method is based on the ILS metaheuristic presented in Chapter 3 for the basic variant of the IRP, it does not correspond to a straightforward extension of the metaheuristic. Adapting it and then reaching a good performance required a considerable effort since the PIRP and the basic IRP have some fundamental differences. First, in addition to delivery and inventory decisions (also present in the basic IRP), in the PIRP the supplier has also to decide the amount of product of each available age that will be used to satisfy the customer demands (consumption decisions). Additionally, in the PIRP we deal with a perishable product that can be available in different ages every time period, therefore the delivery, consumption and inventory decisions are also specific for each available age. Thus, in addition to the components mentioned in the previous sections, we tested many different additional strategies and components to further improve the performance of the method, among which it is worth mentioning the following. First, we tried including the MCF formulation to optimize the continuous amounts of every feasible solution found by the construction heuristic. This resulted in considerably better initial solutions but the gain in solution quality did not pay off the additional computational effort required for solving many LPs, particularly for the largest problem instances. Additionally, using these better initial solutions did not result in better final solutions. Also, in the construction heuristic, when we do not allow `look_ahead` to take positive values, then it fails to find a feasible solution for one of the tested instances. Therefore, we use the construction heuristic as stated in Section 4.4.1.

For the perturbation phase, we developed biased operators and rules for the selection of the components of the solutions that would be perturbed. However, using more sophisticated operators resulted in only marginal gains and, therefore, we opted for setting simple random selection rules.

Finally, for each instance we only construct the MCF formulation once and then we update the right-hand side of constraints (4.74) and (4.75) every time the formulation is solved. It is worth highlighting that this allowed us to obtain a speed-up of five times when compared to a method that constructs the entire MCF formulation every time it is called.

4.5 Computational experiments

In this section we describe the computational experiments performed with the proposed formulations and the hybrid method. All the algorithms were coded in C++ and run on a 2.67 GHz Intel Xeon X5650 Westmere processor with one thread and 36 GB of RAM. We used CPLEX 12.8 as MIP and LP solver. We turned off CPLEX's parallel mode and set the CPLEX MIP tolerance parameter to 10^{-6}. All other CPLEX parameters were set to their default values. The test instances and computational experiments are discussed in the subsequent sections.

4.5.1 Test instances

In our computational experiments we used two sets of problem instances. The first one, proposed by Coelho and Laporte (2014b) (which we will refer to as set CL), is composed of 60 instances divided into 12 subsets with 5 instances each. In addition, we propose a second set of instances, containing 55 (larger) problem instances divided into 11 subsets of 5 instances each. We will refer to this new set as set A. The sizes of the instances of both sets vary in terms of the number of customers (N), the maximum age of the product (S), the number of vehicles (K) and the size of the planning horizon (T). The ranges of these dimensions for both sets are summarized in Table 4.2. Notice that the instances that we propose are larger than those of set CL in terms of the length of the planning horizon, number of vehicles and maximum age of the product, rather than in terms of the number of customers.

Parameter	Set CL	Set A
N	From 10 to 50	From 10 to 40
S	From 2 to 5	From 5 to 10
K	From 1 to 3	From 2 to 8
T	From 3 to 10	From 10 to 20

Table 4.2: Dimensions of the problem instances

In our new instances, the parameters were generated according to the procedure described in Coelho and Laporte (2014b), except for r^t and I^0_{00} (amount made available and initial inventory at the supplier, respectively) given that the authors do not describe how the values of these parameters are generated. Therefore, following the common practice in the IRP literature (Archetti et al., 2007, 2012) we assume that, in each period, the supplier has a supply of products large enough to always be able to serve all its customers. Thus, we set $r^t = 1.5 \sum_{i \in \mathcal{C}} d^1_i,\ \forall t \in \mathcal{T}$ and $I^0_{00} = r^1$. It is worth mentioning that the instances have revenues that are nonincreasing with the age, i.e., $u_{is} \geq u_{i,s+1},\ \forall s \in \mathcal{S} \backslash \{S\}$, and the inventory holding costs are random for both the supplier and the customers (using a formula that includes an increase with the age of the product). Notice that random inventory holding costs represent the most general case for these values. Travel costs correspond to Euclidean distances rounded to the nearest integer. Additionally, in Section 4.5.4, we analyze the results of some further experiments performed with different configurations for the values of revenue, holding and travel costs.

4.5.2 Comparison of the formulations

In this section, we compare the results obtained with the proposed formulations. Recall that we have stated five formulations: arc-based (AB), transportation-based I and II with and without a vehicle index (TP-I, TP-II, TP-I-nk and TP-II-nk, respectively). It is worth remembering that, for these experiments, we did not include any additional valid inequality to the formulations. Experiments including additional valid inequalities are presented in Section 4.5.2.1.

First, in Table 4.3, we report the values of the LP relaxations provided by the formulations and the running times required to solve them. When solving the LP relaxations, in addition to removing the integrality conditions on the variables, we drop the SECs (and GFSECs) of all the formulations. We do this because the separation algorithm that we use for the formulations without a vehicle index is a heuristic procedure for fractional solutions, i.e., it may fail to find a violated inequality. However, as we will show in the upcoming experiments, despite the fact that we use this heuristic procedure, the formulations without a vehicle index had a better performance than the formulations with a vehicle index (which use an exact separation algorithm). In the table, columns 1 and 2 display the instance set and the number of instances in the respective set, respectively. Columns labeled with “Difference to AB” show the relative difference between the values of the LP relaxation of the respective formulations and formulation AB, computed using the formula $100 \times (z^* - z^{AB})/z^{AB}$, where z^* is the LP relaxation value of the formulation and z^{AB} is the value of the LP relaxation of formulation AB. Columns labeled with “Running time” report the time in seconds that the solver required to solve the LP. The results are reported separately in the same table for the two sets of instances (sets CL and A).

Each row represents the average value for all the (five) instances of the respective set (column 1), except for row “All” which shows the average values over all the instances of the respective set (CL and A).

It is possible to observe that the LP relaxations of the formulations without a vehicle index can be considerably worse than the other formulations since, on average, the LP relaxation bounds provided by them are 3.85% and 13.30% greater than the ones of formulation AB, for set CL and A, respectively (with a maximum difference of 24.56%). The LP relaxation of the reformulations with a vehicle index is only slightly worse than the ones of formulation AB (less than 0.5%, for all sets). Notice also that the bounds of formulations TP-I and TP-II (and their respective counterparts without a vehicle index) are identical. In addition, as a result of the considerably larger number of variables in the reformulations with a vehicle index, the times to solve their LP relaxation are large when compared to formulation AB and to the reformulations without a vehicle index. These differences are remarkably larger for those instance sets with the largest values of S and T in set A, given that the number of delivery variables (q) in those formulations grows quickly with increasing values for these parameters.

Instance set		Difference to AB (%)				Running time				
(N-S-K-T)	#	TP-I	TP-I-nk	TP-II	TP-II-nk	AB	TP-I	TP-I-nk	TP-II	TP-II-nk
10-2-1-3	5	0.11	1.72	0.11	1.72	0.00	0.00	0.00	0.00	0.00
10-3-1-6	5	0.26	5.10	0.26	5.10	0.01	0.01	0.01	0.01	0.01
10-5-1-10	5	0.28	4.10	0.28	4.10	0.02	0.03	0.03	0.03	0.03
20-2-2-3	5	0.39	4.55	0.39	4.55	0.01	0.01	0.00	0.02	0.01
20-3-2-6	5	0.25	4.47	0.25	4.47	0.03	0.05	0.01	0.04	0.02
20-5-2-10	5	0.13	4.78	0.13	4.78	0.18	0.33	0.08	0.34	0.07
30-2-2-3	5	0.16	2.06	0.16	2.06	0.01	0.02	0.01	0.02	0.01
30-3-2-6	5	0.26	3.85	0.26	3.85	0.07	0.07	0.04	0.08	0.03
30-5-2-10	5	0.14	4.35	0.14	4.35	0.32	0.48	0.10	0.47	0.10
40-2-3-3	5	0.45	3.42	0.45	3.42	0.06	0.07	0.01	0.06	0.01
40-3-3-6	5	0.26	4.21	0.26	4.21	0.19	0.25	0.04	0.26	0.04
50-2-3-3	5	0.27	3.59	0.27	3.59	0.06	0.09	0.02	0.08	0.02
All		0.25	3.85	0.25	3.85	0.08	0.12	0.03	0.12	0.03
10-7-2-15	5	0.41	11.20	0.41	11.20	0.14	0.33	0.09	0.32	0.09
10-10-2-15	5	0.30	13.72	0.30	13.72	0.19	0.64	0.18	0.75	0.16
10-10-2-20	5	0.36	11.89	0.36	11.89	0.41	1.33	0.34	1.49	0.32
20-7-4-15	5	0.19	11.89	0.19	11.89	1.15	4.18	0.26	3.63	0.22
20-10-4-15	5	0.18	10.78	0.18	10.78	1.52	7.35	0.43	6.52	0.38
20-10-6-15	5	0.17	24.56	0.17	24.56	1.98	11.40	0.37	11.33	0.38
30-7-4-15	5	0.14	8.84	0.14	8.84	2.24	9.75	0.35	8.45	0.36
30-7-8-15	5	0.12	16.71	0.12	16.71	6.29	25.11	0.37	27.07	0.33
30-10-8-15	5	0.15	18.38	0.15	18.38	8.40	52.17	0.67	57.97	0.60
40-5-4-10	5	0.08	7.37	0.08	7.37	1.14	2.85	0.15	2.52	0.13
40-5-8-10	5	0.13	10.93	0.13	10.93	2.87	9.08	0.15	6.97	0.13
All		0.20	13.30	0.20	13.30	2.39	11.29	0.31	11.55	0.28

Table 4.3: Comparison of the LP relaxation of the formulations

Table 4.4 reports the results when we impose a time limit of two hours to solve the MIP formulations. In the table, column “#” indicates the number of instances in each set. Then, for each formulation, column “#O” shows the number of instances of the set solved to optimality, column “#F” shows the number of instances of the set for which a feasible solution was found. In addition, we report the average relative optimality gap (“Opt gap”) of the solutions of the set (as a percentage) and the average total CPU time (“Total time”) in seconds for all the instances of the set. Each row shows the average value for optimality gap and CPU time over all the instances of the set (column 1) for which the respective formulation could find at least a feasible solution (column "#F"). Row "All" displays the sum of the values in the respective columns (for "#", "#O" and "#F"). We do not report summary values for the remaining columns ("Opt gap" and "Total time") to avoid misleading comparisons as not all the formulations could solve to optimality or find feasible solutions for the sames instances. Unfilled cells ("–") indicate that no feasible solution was found for all the instances of the set. The results are reported separately in the same table for the two sets of instances.

It is possible to observe that all the reformulations were able to find a number of feasible solutions that is larger than or equal to the number of feasible solutions found by formulation AB. The differences are especially remarkable for the formulations without a vehicle index (TP-I-nk and TP-II-nk), which were able to find feasible solutions for 67% and 71% more instances than formulation AB, respectively. Additionally, the reformulations with a vehicle index (TP-I and TP-II) found three and two optimal solutions more than formulation AB, respectively, while both reformulations without a vehicle index found eight optimal solutions more than formulation AB (which represent 30% more optimal solutions). These results are also shown in Figure 4.2. It is worth mentioning that when for a given set no optimal solution was found but at least a feasible solution was obtained and the average time is less than 7,200 seconds, it means that the optimizer ran out of memory before reaching the time limit. Specifically, that happened once (one instance of subset 20-5-2-10) and three times (one instance of subset 20-5-2-10, 30-5-2-10 and 10-7-2-15) for the formulation TP-I-nk and TP-II-nk, respectively. Finally, notice that when the instance size grows, the solver starts failing to find feasible solutions within the time limit, especially for the formulations with a vehicle index. As expected, this degradation in the performance is more noticeable when we increase T, S and K than when we increase N. This is because the number of variables and constraints of the formulations increases faster with increasing values of those former parameters.

Table 4.5 summarizes the relative optimality gaps of the solutions but only considering instances for which all the formulations found at least a feasible solution. Column 2 ("#") displays the number of instances of the set (column 1) for which at least a feasible solution was found by all the formulations. The first set of columns ("B&C optimality gap") shows, for each formulation, the relative optimality gaps of the computed feasible solutions using the upper (dual) bound of the corresponding B&C algorithm. "Best dual bound gap" shows the relative difference of the solutions of the respective formulations compared to the best upper (dual) bound among all the formulations, computed as $100 \times (\bar{z} - z)/z$, where $\bar{z}$ is the best dual upper bound computed at the end of the B&C algorithm over all five formulations and z is the objective value of the solution of the model. Each row displays the average gap value over all the instances of the set (column 1) for which all the formulations found at least a feasible solution (column "#"), except for row "All" which shows the average values over all the 57 instances for which all the formulations found at least a feasible solution. Notice that the results in the first 11 rows (sets 10-2-1-3 to 50-2-3-3) correspond to those of set CL and the last two rows (sets 10-7-2-15 and 10-10-2-15) are from set A. The results show that, despite the fact that the formulations without a vehicle index can have a considerably worse LP relaxation, better

Instance set		AB				TP-I				TP-I-nk				TP-II				TP-II-nk			
(*N*-*S*-*K*-*T*)	#	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time
10-2-1-3	5	5	5	0.00	0.05	5	5	0.00	0.04	5	5	0.00	0.32	5	5	0.00	0.04	5	5	0.00	0.35
10-3-1-6	5	5	5	0.00	0.29	5	5	0.00	0.40	5	5	0.00	1.01	5	5	0.00	0.37	5	5	0.00	0.51
10-5-1-10	5	5	5	0.00	3.00	5	5	0.00	3.19	5	5	0.00	4.33	5	5	0.00	3.32	5	5	0.00	4.71
20-2-2-3	5	5	5	0.00	43.96	5	5	0.00	21.85	5	5	0.00	36.08	5	5	0.00	25.11	5	5	0.00	21.83
20-3-2-6	5	1	5	1.14	6,137.72	3	5	0.47	4,205.82	4	5	0.43	1,906.76	2	5	0.67	5,044.31	4	5	0.39	1,918.78
20-5-2-10	5	0	5	7.37	7,200.02	0	5	6.88	7,200.02	0	5	3.00	7,036.44	0	4	5.56	7,200.03	0	5	2.33	6,964.66
30-2-2-3	5	4	5	0.23	2,295.86	5	5	0.00	579.75	4	5	0.05	1,461.97	5	5	0.00	868.97	4	5	0.02	1,455.03
30-3-2-6	5	1	5	1.43	5,845.56	1	5	1.22	5,877.06	2	5	0.90	4,955.59	1	5	1.28	5,857.42	2	5	0.73	5,339.87
30-5-2-10	5	0	3	4.02	7,200.03	0	3	4.21	7,200.05	0	5	1.22	7,200.03	0	3	1.51	7,200.07	0	5	1.13	6,964.58
40-2-3-3	5	0	5	3.47	7,200.02	0	5	2.73	7,200.02	3	5	0.18	3,727.92	0	5	2.33	7,200.01	3	5	0.33	4,007.69
40-3-3-6	5	0	0	–	–	0	1	11.50	7,200.02	0	5	3.71	7,200.04	0	0	–	–	0	5	2.97	7,200.02
50-2-3-3	5	0	5	2.78	7,200.02	0	5	2.25	7,200.01	1	5	1.07	5,770.43	0	5	2.63	7,200.01	1	5	0.91	5,805.27
All	60	26	53			29	54			34	60			28	52			34	60		
10-7-2-15	5	0	4	14.34	7,200.01	0	5	11.81	7,200.03	0	5	4.82	7,200.05	0	5	16.94	7,200.02	0	5	5.23	6,935.15
10-10-2-15	5	0	2	9.99	7,200.01	0	4	22.71	7,200.04	0	5	6.51	7,200.07	0	2	18.95	7,200.05	0	5	6.45	7,200.03
10-10-2-20	5	0	0	–	–	0	0	–	–	0	5	13.25	7,200.05	0	0	–	–	0	5	10.13	7,200.07
20-7-4-15	5	0	0	–	–	0	0	–	–	0	5	14.16	7,200.06	0	0	–	–	0	5	13.90	7,200.06
20-10-4-15	5	0	0	–	–	0	0	–	–	0	2	10.91	7,200.04	0	0	–	–	0	2	10.26	7,200.02
20-10-6-15	5	0	0	–	–	0	0	–	–	0	3	10.62	7,200.04	0	0	–	–	0	3	11.44	7,200.02
30-7-4-15	5	0	0	–	–	0	0	–	–	0	2	11.25	7,200.15	0	0	–	–	0	3	14.54	7,200.03
30-7-8-15	5	0	0	–	–	0	0	–	–	0	3	18.83	7,200.07	0	0	–	–	0	4	18.63	7,200.06
30-10-8-15	5	0	0	–	–	0	0	–	–	0	1	15.51	7,200.11	0	0	–	–	0	0	–	–
40-5-4-10	5	0	0	–	–	0	0	–	–	0	3	13.55	7,200.05	0	0	–	–	0	5	15.70	7,200.06
40-5-8-10	5	0	0	–	–	0	0	–	–	0	5	15.23	7,200.02	0	0	–	–	0	4	13.41	7,200.07
All	55	0	6			0	9			0	39			0	7			0	41		

Table 4.4: Comparison of the MIP results obtained with the formulations

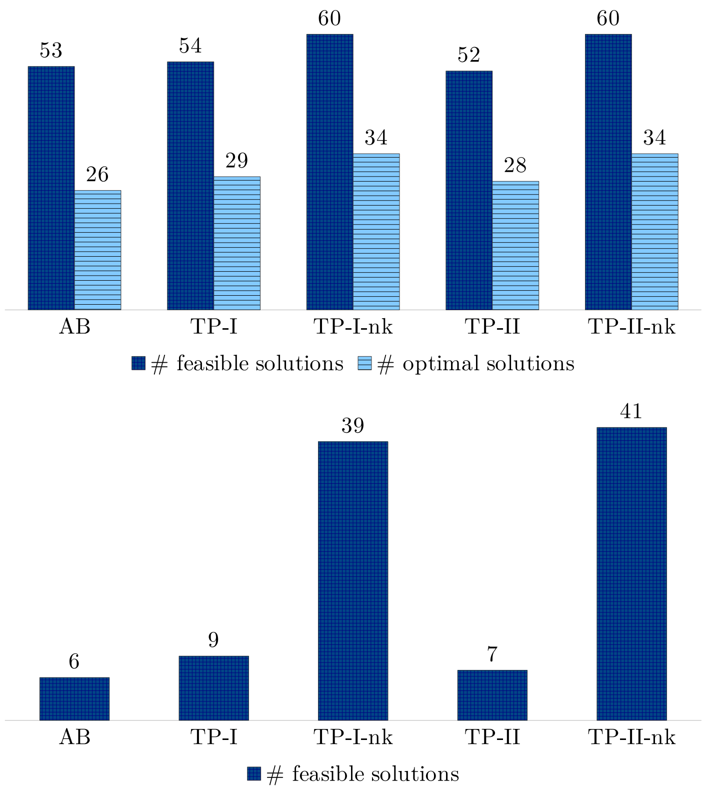


Figure 4.2: Results with the different formulations for sets CL (top) and A (bottom)

optimality gaps can be obtained at the end of their B&C algorithms. These formulations can provide solutions of better quality for the largest instances of the table, as shown by the values of the gaps to the best dual bounds. Notice that for sets 10-7-2-15 and 10-10-2-15 (which are part of set A), the formulations without a vehicle index (TP-I-nk and TP-II-nk) found solutions with gaps to the best dual bounds significantly better than the ones provided by the other three formulations. Also, the results highlight the sensitivity of the formulations to increases in the values of S, K and T.

Instance set		B&C optimality gap (%)					Best dual bound gap (%)				
(N-S-K-T)	#	AB	TP-I	TP-I-nk	TP-II	TP-II-nk	AB	TP-I	TP-I-nk	TP-II	TP-II-nk
10-2-1-3	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10-3-1-6	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10-5-1-10	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20-2-2-3	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20-3-2-6	5	1.14	0.47	0.43	0.67	0.39	0.46	0.33	0.41	0.41	0.39
20-5-2-10	4	5.75	5.58	2.88	5.56	2.29	4.36	4.37	2.52	4.31	2.27
30-2-2-3	5	0.23	0.00	0.05	0.00	0.02	0.01	0.00	0.01	0.00	0.00
30-3-2-6	5	1.43	1.22	0.90	1.28	0.73	0.86	0.74	0.89	0.71	0.73
30-5-2-10	2	3.71	5.35	0.75	1.30	0.51	3.14	4.86	0.75	0.82	0.50
40-2-3-3	5	3.47	2.73	0.18	2.33	0.33	1.91	1.31	0.18	0.92	0.33
50-2-3-3	5	2.78	2.25	1.07	2.63	0.91	1.97	1.43	1.05	1.89	0.90
10-7-2-15	4	14.34	12.49	4.59	17.20	4.76	9.06	7.53	4.56	11.96	4.66
10-10-2-15	2	9.99	10.51	3.40	18.95	3.44	6.43	7.00	3.31	14.94	3.40
All	57	2.68	2.41	0.90	2.91	0.84	1.74	1.58	0.86	2.04	0.83

Table 4.5: Results for instances with feasible solutions found by all the formulations

Finally, Table 4.6 shows the results for only those instances that were solved to optimality within the time limit by all the formulations. Column 2 ("#") displays the number of instances of the set (column 1) for which all formulations found the optimal solution. Each row shows the average time over all the instances of the set (column 1), except for row "All" which shows the average values over all the 25 instances solved to optimality within the time limit by all the formulations. All the results are from set CL. The results reveal the speed-up obtained with the reformulations, especially the ones without a vehicle index. In particular, formulations TP-I-nk and TP-II-nk use, in total, only 6.85% and 4.93% of the time required by formulation AB, respectively. The results also reveal that a significant portion of the total CPU time ("Total time") is spent proving the optimality of the solutions since in most of the cases the optimal solutions are found early ("Time to best") in the B&C tree.

		AB		TP-I		TP-I-nk		TP-II		TP-II-nk	
Instance set (N-S-K-T)	#	Total time	Time to best	Total time	Time to best	Total time	Time to best	Total time	Time to best	Total time	Time to best
10-2-1-3	5	0.05	0.05	0.04	0.04	0.32	0.31	0.04	0.04	0.35	0.33
10-3-1-6	5	0.29	0.24	0.40	0.35	1.01	0.98	0.37	0.32	0.51	0.45
10-5-1-10	5	3.00	2.62	3.19	2.55	4.33	4.05	3.32	2.66	4.71	4.48
20-2-2-3	5	43.96	23.34	21.85	12.82	36.08	28.02	25.11	13.57	21.83	17.32
20-3-2-6	1	1,888.53	213.08	753.52	51.25	12.09	11.41	1,984.50	77.44	10.66	10.34
30-2-2-3	3	368.49	151.97	61.23	28.75	5.80	4.40	85.25	45.30	6.24	5.49
30-3-2-6	1	427.71	182.28	585.22	304.36	12.19	11.73	487.02	58.48	13.84	13.75
All	25	146.33	39.30	65.99	20.83	10.02	8.13	114.86	14.19	7.21	6.14

Table 4.6: Results for instances with optimal solutions found by all the formulations

4.5.2.1 Impact of the valid inequalities

The purpose of this section is to analyze the results obtained when we strengthen the formulations by including the valid inequalities of Section 4.3.7. Table 4.7 shows the number of optimal and feasible solutions found when the different valid inequalities are included in the formulations. We considered the instance set CL only since we can find a considerably large number of feasible solutions for these instances. Again, we set a time limit of two hours to solve each instance by each formulation. In the cells of columns 2-6, we show the number of optimal and feasible solutions, respectively (separated by a comma), found by the formulation specified in the respective header when including the inequalities given in column 1. "Base case" shows the results when no valid inequality is considered in the formulation and "All" shows the results when we include all the inequalities simultaneously. Unfilled cells ("–") indicate that the inequalities were not applied to the formulations (symmetry breaking constraints to the formulations without a vehicle index).

The results show that none of the valid inequalities was able to improve alone the performance of all the formulations in terms of both the number of optimal and feasible solutions found within the time limit. Only inequality (4.63) was able to provide a number of optimal and feasible solutions that is greater than or equal to the base case for all the formulations. This could be explained by the fact that CPLEX already includes some general valid inequalities (in addition to the preprocessing operations that the solver applies to the formulation), which does not allow to reveal potential gains obtained by including the given valid inequalities. Note that some of the inequalities are able to improve the performance of some of the formulations in either the number of optimal or feasible solutions, e.g., (4.69) for the number of feasible solutions, but not in both at the same time. It is worth mentioning that we tried including combinations of these inequalities simultaneously, but no performance improvement was obtained.

	Formulation				
	AB	TP-I	TP-I-nk	TP-II	TP-II-nk
Base case	26, 53	29, 54	34, 60	28, 52	34, 60
(4.63)	26, 54	29, 54	34, 60	29, 54	34, 60
(4.64)	24, 54	27, 54	33, 60	27, 53	32, 60
(4.65)	26, 55	29, 53	–	29, 52	–
(4.66)	26, 54	29, 53	–	28, 53	–
(4.67)	26, 53	29, 53	32, 60	28, 56	32, 60
(4.68)	25, 53	28, 52	33, 60	29, 53	33, 60
(4.69)	25, 56	27, 56	32, 60	29, 54	32, 60
(4.63)-(4.66)	26, 55	29, 52	33, 60	29, 53	32, 60
(4.67)-(4.69)	26, 54	28, 54	34, 60	29, 52	33, 60
All	24, 56	29, 57	32, 60	28, 57	33, 60

Table 4.7: Number of optimal and feasible solutions found by the formulations when including the valid inequalities

The impact of the valid inequalities on the CPU times are presented in Table 4.8. In the table, for each formulation, column "#O" shows the number of instances of the set solved to optimality, column "#F" shows the number of instances of the set for which a feasible solution was found, "Opt gap" shows the average relative optimality gap of the solutions of the set (as a percentage) and "Total time" displays the average total execution time in seconds for all the instances in the set. Recall that each set contains five instances.

The results show that there is no consistency in the impact of the valid inequalities over all five formulations and all the sets of instances when compared to the results without any valid inequality. For some sets, the valid inequalities improved the performance of some formulations (e.g., formulation AB on set 20-5-2-10) but in other cases, they worsened it (e.g., formulation TP-I-nk on set 30-2-2-3). The number of feasible solutions found by all the formulations without a vehicle index increased, while the number of optimal solutions was reduced for some of the formulations. We believe this may be in part due to the fact that CPLEX already generates some general valid inequalities that do not allow to reveal potential gains obtained by including the given valid inequalities.

4.5.3 Results with the optimization-based ILS

Next, we analyze the performance of the optimization-based ILS hybrid method. In all tables of this section, column "Best sol gap" shows the relative difference of the solutions obtained with our heuristic method (z^h) to the best feasible solution found by all the MIP formulations (z^f), computed as $100 \times (z^f - z^h)/z^f$; "Opt gap" shows the relative optimality gap of the obtained solutions, computed using the formula $100 \times (\bar{z} - z^h)/\bar{z}$, where $\bar{z}$ is the best dual upper bound

Instance set	AB				TP-I				TP-I-nk				TP-II				TP-II-nk			
(N-S-K-T)	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time
10-2-1-3	5	5	0.00	0.03	5	5	0.00	0.02	5	5	0.00%	0.03	5	5	0.00	0.03	5	5	0.00	0.03
10-3-1-6	5	5	0.00	0.21	5	5	0.00	0.22	5	5	0.00	0.48	5	5	0.00	0.23	5	5	0.00	0.56
10-5-1-10	5	5	0.00	1.75	5	5	0.00	2.08	5	5	0.00	4.77	5	5	0.00	1.75	5	5	0.00	4.38
20-2-2-3	5	5	0.00	26.21	5	5	0.00	15.36	5	5	0.00	29.41	5	5	0.00	16.45	5	5	0.00	22.65
20-3-2-6	1	5	1.10	6,724.99	3	5	0.43	3,740.16	3	5	0.59	3,185.49	3	5	0.55	3,464.04	4	5	0.38	1,904.10
20-5-2-10	0	5	5.34	7,200.01	0	5	5.20	7,200.03	0	5	2.75	7,200.02	0	5	8.60	7,200.02	0	5	2.71	6,817.40
30-2-2-3	2	5	0.52	4,345.10	5	5	0.00	1,240.25	4	5	0.02	1,465.08	4	5	0.03	1,490.79	4	5	0.04	1,468.55
30-3-2-6	1	5	1.48	5,976.06	1	5	1.48	5,842.84	1	5	0.88	5,762.64	1	5	1.53	5,799.09	2	5	0.94	5,270.90
30-5-2-10	0	5	9.21	7,200.03	0	4	2.36	7,200.04	0	5	1.47	6,899.24	0	4	4.44	7,200.03	0	5	2.09	6,872.01
40-2-3-3	0	5	3.16	7,200.01	0	5	2.61	7,200.01	3	5	0.20	3,462.04	0	5	2.56	7,200.01	2	5	0.26	4,649.26
40-3-3-6	0	1	7.39	7,200.06	0	3	13.01	7,200.03	0	5	3.81	7,200.01	0	3	10.92	7,200.04	0	5	5.44	7,200.03
50-2-3-3	0	5	2.97	7,200.01	0	5	2.87	7,200.01	1	5	1.12	5,915.02	0	5	3.14	7,200.02	1	5	1.02	5,781.86
All	24	56			29	57			32	60			28	57			33	60		

Table 4.8: Results of the formulations including all the valid inequalities

computed at the end of the B&C algorithm over all five formulations (Section 4.5.2); “Total time” displays the total time (in seconds) required by the algorithm; and “Imp” shows the relative improvement in the objective function value (profit) obtained by including the SI formulation in the method, computed as $100 \times (z^{h_2} - z^{h_1})/z^{h_1}$, where z^{h_1} and z^{h_2} are the solutions found by the heuristic before and after applying the SI formulation in the method, respectively. Note that negative values of “Best sol gap” indicate that our method found a feasible solution that is better than the best solution provided by the five formulations since we are maximizing profit.

Tables 4.9 and 4.10 show the results obtained with the optimization-based ILS. The stopping criterion was the number of iterations, which we set to 500. The value of `max_perturb` was set to $\lceil 0.07 \times N \rceil$. This value was determined through empirical preliminary experiments using a random subset of the problem instances. Recall that this parameter defines the number of elements of a solution that will be changed in each call of the perturbation mechanism. For the SI formulation, we defined a time limit of 60 seconds and set β to 1, i.e., we allow at most one removal or insertion per route. We present both results, with and without the SI formulation. We run the algorithm only once for each instance given that, as will be shown in Section 4.5.3.1, the results are relatively consistent between different runs. We separated the results into two tables. In the tables, each row displays the average result of the five instances of the given set (column 1), except for “Best sol gap”, whose values were calculated over the instances for which we could find a feasible solution with at least one of the formulations (column “#”). The last row (“All”) shows the average results over all the instances considered. It is worth mentioning that for all the instances of set CL at least one formulation was able to find a feasible solution, while for only 47 instances of set A a formulation could find a feasible solution.

The results on the instance set CL (Table 4.9) show that the proposed method is able to find high-quality solutions in relatively short running times. Specifically, solutions with an average optimality gap of 2.08% (1.87%) were obtained within 5.96 (6.53) seconds, on average, for all the instances of this set without (with) the SI formulation. Note that the average CPU time on this data set using the B&C algorithm with the TP-I-nk (TP-II-nk) formulation is 3,275 (3,306) seconds. The results also reveal the sensitivity of the method to an increase in the number of periods (T), mainly given that the effort required in the local search phase of the method, as well as the MCF formulation size, depend primarily on T. The inclusion of the SI formulation as a post-optimization phase provided an average relative improvement of the objective value (profit) of 0.20% with an increase of 9,56% in the running time. The SI formulation improved 43 (out of 60) solutions in this set. It is also worth mentioning that in 45 of the instances of this set, the MCF problem formulation led to the best feasible solution (before applying the SI formulation) in either an outer iteration after the local search phase or an inner iteration after a perturbation operator. These results highlight the importance of the two mathematical programming components for the hybrid method for these instances.

The results on the new problem instance set A (Table 4.10) show the impact of solving larger instances since, as expected, significantly larger running times are required to perform 500 iterations. Also, larger optimality gaps are obtained for the solutions on these instances. This fact does not necessarily reflect a degradation in the quality of the solutions that our method

Instance set		Without SI			With SI			
(N-S-K-T)	#	Best sol gap (%)	Opt gap (%)	Total time	Best sol gap (%)	Opt gap (%)	Total time	Imp (%)
10-2-1-3	5	0.02	0.02	0.19	0.02	0.02	0.20	0.00
10-3-1-6	5	0.12	0.12	0.35	0.11	0.11	0.39	0.01
10-5-1-10	5	0.59	0.59	0.74	0.47	0.47	0.93	0.12
20-2-2-3	5	1.20	1.20	0.60	1.14	1.14	0.65	0.06
20-3-2-6	5	1.74	2.04	1.36	1.43	1.73	1.61	0.31
20-5-2-10	5	1.01	3.14	14.02	0.59	2.73	16.05	0.42
30-2-2-3	5	0.42	0.42	1.14	0.37	0.37	1.21	0.06
30-3-2-6	5	1.68	2.16	2.71	1.53	2.01	3.09	0.16
30-5-2-10	5	1.30	2.16	25.95	0.96	1.82	28.20	0.35
40-2-3-3	5	4.72	4.90	2.08	4.46	4.64	2.29	0.28
40-3-3-6	5	1.94	3.76	19.28	1.50	3.39	20.38	0.39
50-2-3-3	5	4.54	5.13	3.12	4.28	4.87	3.34	0.27
All		1.60	2.14	5.96	1.40	1.94	6.53	0.20

Table 4.9: Results with the optimization-based ILS on instance set CL

can provide for these instances, but can reflect the low quality of the dual upper bounds provided by the formulations on these instances. Recall that, using the B&C algorithms, even finding feasible solutions was difficult for these instances. On average, the hybrid method finds solutions with objective values 0.45% (1.73%) better than the best solutions found by all the formulations without (with) the SI formulation, for the instances for which the formulations could provide a feasible solution (47 instances). An average relative improvement in the objective value (profit) of 1.07% was obtained when the SI formulation was included in the method at a cost of an increase of around one minute in the total running time (from 107 seconds to 166 seconds, which represents a 56% time increase). Note that the average CPU time of the B&C algorithm with the TP-I-nk (TP-II-nk) formulation (for those instances of set A for which a feasible solution was found) was 7,200 (7,167) seconds. For the instances of this set, the SI formulation improved 32 (out of 55) of the solutions. Also, the MCF problem formulation led to 45 of the best feasible solutions for the instances of this set (before applying the SI formulation). Notice also that, on average, for most of the instances of this set the solver reached the time limit (60 seconds) for solving the SI formulation. The latter fact shows the difficulty to solve this formulation and can explain why for some sets the SI formulation could not find a better solution. It is worth mentioning that Archetti et al. (2012) proved that the customer assignment problem described by the SI formulation is NP-hard for the single-vehicle case.

4.5.3.1 Evaluation of the hybrid method with different configurations

The purpose of this section is to assess the performance of the method under different configurations. First, Table 4.11 shows the results of the method when we change the number of iterations used as stopping criterion (column 1). The results show that, for set CL, even for a small number of iterations (e.g., 100), the method is able to find relatively good quality solutions in very short running times. For set A, from 250 iterations onwards the method is able to find solutions with an average objective function value that is better than the objective function value of the best solutions provided by all the formulations within two hours, which further shows the ability of the method to find good feasible solutions in relatively short CPU times.

Instance set (N-S-K-T)	#	Without SI Best sol gap (%)	Without SI Opt gap (%)	Without SI Total time	With SI Best sol gap (%)	With SI Opt gap (%)	With SI Total time	With SI Imp (%)
10-7-2-15	5	4.09	8.12	13.69	1.88	5.99	71.50	2.40
10-10-2-15	5	3.63	8.99	16.60	1.61	7.10	76.62	2.09
10-10-2-20	5	2.95	10.96	22.54	0.67	8.85	82.56	2.41
20-7-4-15	5	-1.03	10.83	53.69	-2.51	9.51	113.73	1.47
20-10-4-15	2	-0.69	8.81	81.20	-1.00	8.71	141.27	0.12
20-10-6-15	3	2.66	15.28	124.06	2.66	15.28	184.20	0.00
30-7-4-15	4	-4.50	7.90	113.95	-5.79	6.80	174.02	1.19
30-7-8-15	4	-0.78	14.17	236.38	-0.78	14.17	296.52	0.00
30-10-8-15	1	-0.41	14.24	334.36	-0.41	14.24	394.67	0.00
40-5-4-10	5	-8.11	6.32	66.67	-9.13	5.44	126.70	0.94
40-5-8-10	5	-2.48	10.04	109.10	-3.64	9.01	169.15	1.14
All		-0.45	10.51	106.57	-1.73	9.56	166.45	1.07

Table 4.10: Results with the optimization-based ILS on instance set A

# of iter	Set CL Best sol gap (%)	Set CL Opt gap (%)	Set CL Total time	Set A Best sol gap (%)	Set A Opt gap (%)	Set A Total time
100	2.32	2.87	1.83	0.87	12.15	79.95
250	1.69	2.24	3.83	-0.57	10.71	110.61
500	1.32	1.90	6.39	-1.86	9.53	159.31
1000	1.14	1.70	12.37	-2.72	8.72	262.79
2000	1.02	1.57	27.46	-3.62	7.79	497.17

Table 4.11: Results of increasing number of iterations for the hybrid method

To evaluate the impact of the randomness in our method, we executed it five times and computed the percent coefficient of variation (%CV) of the results. The experiment resulted in a %CV of 0.57% for the total profit considering both sets of instances. This result highlights the consistency of the results obtained in different runs of the method. It is worth mentioning that the %CV for set CL is 0.37% while for set A it was 0.78%. Although these values are both very small, the difference between them reveals the impact of the size of the instances of set A as, on average, the results obtained for these instances are (slightly) more dispersed than the results for instances of set CL.

4.5.3.2 Solving the basic variant of the IRP using the optimization-based ILS

In this section we present the results obtained with the optimization-based ILS hybrid method when applied to the basic variant of the IRP. The method was tested on the set of instances proposed in Archetti et al. (2007) for the basic IRP. This set is composed of 160 instances with up to 50 customers and up to 6 time periods. These instances were originally created for the single-vehicle IRP, but they have also been adapted for the case with multiple vehicles by dividing the vehicle capacity by the number of vehicles. For the multiple vehicle case, we compared the optimization-based ILS hybrid heuristic with the hybrid heuristic of Archetti et al. (2017a) (that combines a tabu search and mathematical programming formulations) and the decomposition matheuristic of Chitsaz et al. (2019) (proposed to solve the assembly routing problem and adapted to the IRP). For the single-vehicle case we compared with the hybrid heuristic of Archetti et al. (2012) and also the decomposition matheuristic of Chitsaz et al. (2019).

To solve the basic variant of the IRP (which can be seen as a particular case of the PIRP), we used the method as described in Section 4.4 as a general solution framework and adapted it as follows. First of all, there is the fact that without loss of generality in the solutions of the basic IRP the customers consume the delivered quantities following a first-in, first-out (FIFO) rule (Desaulniers et al., 2016; Archetti and Speranza, 2016). Thus, the relations between delivery, consumption and inventory in the basic IRP are simpler than those of the PIRP, in which the optimal consumption policy does not necessarily follow a FIFO rule. Therefore, when applying the optimization-based ILS hybrid method for the basic variant of the IRP, we can use additional operators in the local search phase (increase/reduce deliveries, merge, transfer and insert visits) without having to optimally define the delivery, consumption and inventory variables every time. Moreover, given that for the basic IRP we do not solve an LP at each inner iteration of the local search phase, it resulted in a significant time speed-up of the method. Taking advantage of this, we applied the local search heuristic in a multi-start fashion, as in Chapter 3. We set to 2,000 outer iterations the stopping criterion of the optimization-based ILS hybrid method when applying it to the basic variant of the IRP.

In Figure 4.3 we show the results when comparing to the methods of Archetti et al. (2017a) and Chitsaz et al. (2019) for the multi-vehicle case. In the figure, we show the average relative difference of the solutions obtained with our heuristic method (z^h) with respect to their solutions (z^m), computed as $100 \times (z^h - z^m)/z^m$ (negative values show better solutions), and the number of instances for which our method found a solution with an objective function value better than those found by their methods. The results are separated according to the number of vehicles. No results are displayed for the case of a single vehicle when comparing to Archetti et al. (2017a) because the authors did not apply their method to these instances. The values of the relative difference were computed over the 160 instances for each number of vehicles, except for the case with five vehicles which has 158 instances because two of the original instances become infeasible in this case. In the figure, it is possible to observe that the optimization-based ILS hybrid method provides reasonably good feasible solutions when compared to state-of-the-art methods, developed either specifically for the multi-vehicle IRP such as the method of Archetti et al. (2017a) or for more general problems such as the method of Chitsaz et al. (2019). Notice that the relative differences tend to improve as the number of vehicles increases.

When compared to the hybrid heuristic of Archetti et al. (2012) for the single-vehicle case, we found solutions with objective function values that are 1.60% higher (average over all the 160 instances) than their solutions. On average, the CPU time of our method was 8.78 seconds for each instance (with a maximum of 41.34 seconds) considering all the instances (multi- and single-vehicle cases). The method of Archetti et al. (2017a) required an average time of more than 1,000 seconds, the method of Chitsaz et al. (2019) used more than 60 seconds on average, while the method of Archetti et al. (2012) can take more than 3,000 seconds. Although these CPU times are not directly comparable due to differences in computational environments, these results highlight the effectiveness of our optimization-based ILS hybrid method. All the results can be found online at `http://chairelogistique.hec.ca/en/scientific-data/`.

When compared to the ILS and SA metaheuristics presented in Chapter 3, the hybrid method

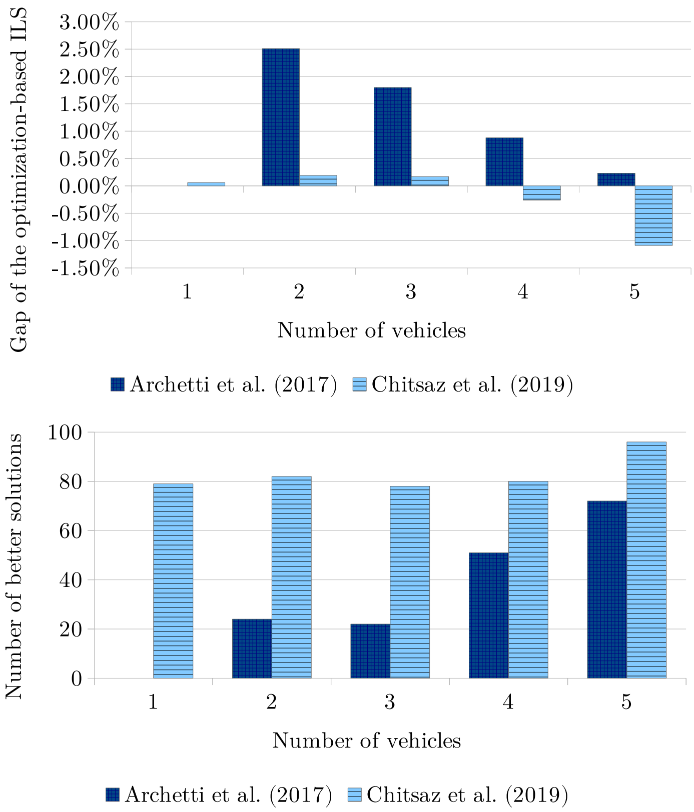


Figure 4.3: Results with the optimization-based ILS hybrid method applied to the basic IRP

found solutions with an average relative difference of 0.90% and 1.37%, respectively, considering all the instances of Archetti et al. (2007). In total, the hybrid method found 279 and 173 solutions that were better than those found by ILS and SA, respectively.

4.5.4 Experiments changing the parameters of the instances

To verify the impact of different scenarios on the solution structure, we perform additional experiments changing the values of several parameters of the problem instances. In these experiments, we used the first four subsets from set CL (10-2-1-3, 10-3-1-6, 10-5-1-10 and 20-2-2-3) given that these are the only sets in which all instances can be solved to optimality within the time limit of two hours by all the formulations. The results of the experiments are shown in Table 4.12 and in Figure 4.4. "Base case" shows the results with the original values of the parameters; "TC" shows the results when we increase the travel cost of each edge of the graph by 10 ($c_{ij} = 10c_{ij},\ \forall (i,j) \in \mathcal{E}$); "SR" shows the results when we set the same revenue for all the ages, using the formula $u_{is} = u_{i0},\ \forall i \in \mathcal{C}, s \in \mathcal{S}$. Notice that the revenues may still be different between customers. "SH" shows the results when we set the same value for the holding costs of all facilities and ages ($h_{is} = 0.5,\ \forall i \in \mathcal{N}, s \in \mathcal{S}$); "ZH" presents the results when we set the holding costs to zero for all facilities and ages ($h_{is} = 0,\ \forall i \in \mathcal{N}, s \in \mathcal{S}$); "DA" shows the results when we double the maximum age of the product ($S = 2S$), setting the corresponding revenue values as $u_{is} = 0.9u_{is-1},\ \forall i \in \mathcal{C}, s = S^0 + 1, \ldots, S$, where S^0 is the previous maximum age, and the inventory holding cost randomly (as the original values); "RSC" displays the results when we reduce by 20% the storage capacity of the customers ($C_i = 0.8C_i,\ \forall i \in \mathcal{C}$). Finally, "RVC" shows the results when we reduce the vehicle capacities using a predefined filling coefficient (Salavati-Khoshghalb et al., 2019). The filling coefficient $\bar{f} = \sum_{i\in\mathcal{C}} \sum_{t\in\mathcal{T}} d_i^t/(T \times K \times Q)$ controls the problem tightness in terms of vehicle capacity. A filling coefficient of 90% was considered. Then, the capacity Q of each vehicle was directly computed from the specified $\bar{f}$. It is worth mentioning that in the scenario "RSC", the initial inventory of some customers may be greater than their new storage capacity (thus making the instances infeasible). In those cases, we set ($I_{i0}^0 = C_i$).

For all the changes performed to the problem instances, we report in Table 4.12 the total profit ("Profit"), which corresponds to the objective function value, the revenue ("Revenue") and the total routing ("Routing") and inventory holding cost ("Inventory") of the optimal solutions. Each cell shows the average for all the instances considered (20 instances). It is worth remembering that the original revenue values are nonincreasing with an increasing age ($u_{is} \geq u_{i,s+1},\ \forall s \in \mathcal{S}\backslash\{S\}$) and the holding costs are random. The impact of the changes on the CPU times of the solver is shown in Section 4.5.4.1. Moreover, in Figures 4.4 and 4.5 we show the average total number of routes (# of routes) and visits (# of visits), the average inventory level at the customers (Avg. I), the average delivery size (Avg. q), the average vehicle fleet usage (Avg. fleet usage), the average vehicle capacity usage (Avg. veh. usage), and the average customers storage capacity usage (Avg. cap. usage) of the solutions, considering the 20 instances used for this experiment.

	Profit	Revenue	Routing	Inventory
Base case	42,410.43	87,966.35	12,368.95	33,186.97
TC	-29,803.19	75,466.60	69,776.00	35,493.79
SR	63,869.51	105,350.40	7,541.15	33,939.74
SH	48,678.03	87,897.85	12,156.15	27,063.68
ZH	75,792.15	87,664.50	11,872.35	0.00
DA	28,533.02	86,964.30	11,677.95	46,753.33
RSC	41,429.38	86,855.90	12,251.40	33,175.12
RVC	41,588.82	88,052.90	12,654.85	33,809.23

Table 4.12: Results when the parameters of the problem instances are changed

The results show that when we increase the travel costs (TC) (compared to the base case), the total revenue is reduced given that the sales revenue values do not pay off the cost of delivering fresh products and consequently the demands tend to be satisfied using older products, which provide less revenue per consumed unit. Analogously, a slight increase in the total inventory cost was observed since extra deliveries, which provide savings in the inventory cost, are not performed anymore given that they are no longer profitable. As expected, the number of routes, visits (with larger deliveries), as well as the vehicle fleet usage of the solutions decrease significantly. In this scenario, the total routing cost increases substantially, compared to the base case, resulting in negative profits. On the other hand, when we set the same revenue for all the ages (SR), the total revenue increases (compared to the base case) given that the income of the sales revenue when satisfying the demand is always maximum. In this scenario, the total revenue becomes constant because the total demand must be satisfied. Additionally, a reduction in the number of routes and visits (with larger delivery quantities) is observed, when compared to the base case, given that in this scenario it is not necessary to perform deliveries in each period to obtain the maximum revenue and only when it is necessary to satisfy a demand or when it can provide savings in the total inventory holding cost. The latter fact results in reduced fleet and vehicle capacity usage, on average. When the holding costs are set to a constant value (SH) it is possible to observe a slight reduction in the average routing cost (as well as in the number of routes, visits and vehicle capacity usage) when compared to the base case. This can be explained by the fact that deliveries that provided savings in the inventory cost are no longer performed. Again, analogously to the total revenue in scenario SR, the total inventory cost becomes a constant term in scenario SH.

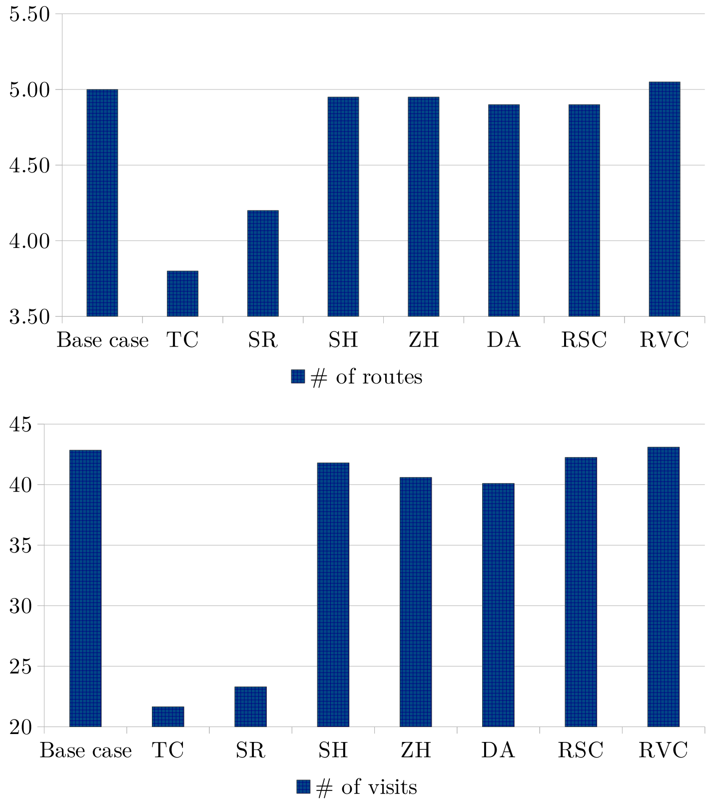


Figure 4.4: Behavior of the solutions for the changes applied (a)

When we compare the scenario without any holding costs (ZH) with the base case, it is possible to observe a slight reduction in the total routing cost (also in the number of routes and visits) given that no savings can be achieved at the level of the holding costs, and we only have a trade-off between transportation costs and revenue values. This results in a reduction of the average delivery quantity (of fresh product, in general), used to maximize the total revenue. Also, reductions in the inventory levels and the different capacity usage measures are observed. For scenario DA, in which we double the maximum age of the product, it is possible to observe a considerable increase in the average inventory level (consequently in the total holding costs and storage capacity usage) as a result of a longer shelf-life. Also, the total revenue decreases as now older products are used to fulfill the demands with the aim of reducing the inventory

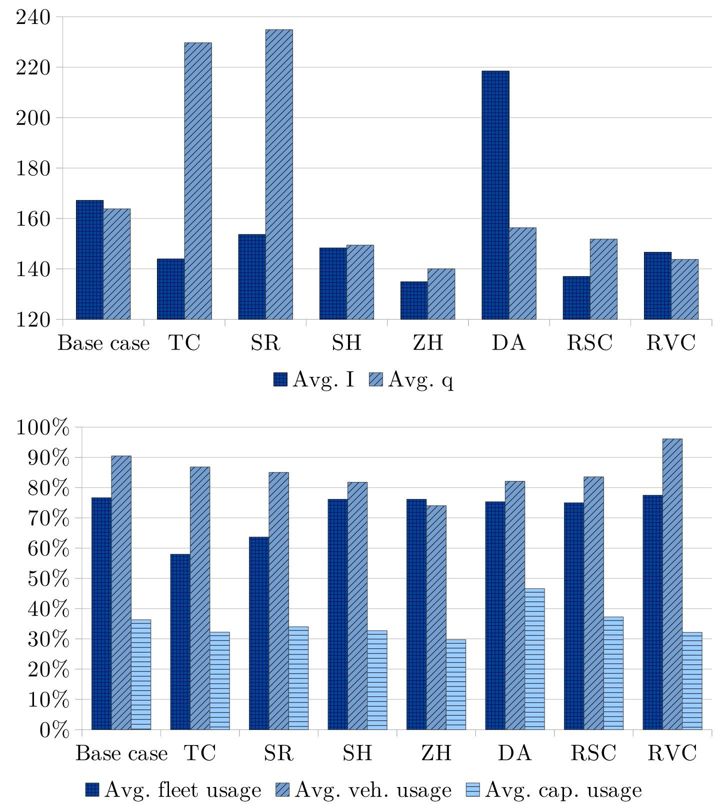


Figure 4.5: Behavior of the solutions for the changes applied (b)

holding costs. This generates a reduction in the delivery quantities, which at the same time results in a reduced number of routes, visits and total transportation costs, as well as vehicle capacity usage. For the scenario with reduced storage capacity (RSC), as expected, we observe a reduction in the average delivery quantity and inventory level (compared to the base case) given that the storage capacity limits the amount that can be delivered in each visit. This results in a reduction of the total revenue and profit. Finally, for the scenario with reduced vehicle capacity (RVC) compared with the base case, we observe a slight increase in the number of routes and visits (with smaller deliveries), resulting in larger routing cost and vehicle capacity usage. Moreover, the total inventory cost increases since deliveries that provide savings in the inventory cost are not performed anymore, given that the vehicle capacity does not allow it and using an additional vehicle does not pay off the savings. This results in a reduction of the total profit of the solutions.

4.5.4.1 CPU times of the solver when changing the parameters of the instances

We report the CPU times required by the solver to solve to optimality all the instances considering the different changes to the instance parameters. It is worth remembering that we used all the instances of subsets 10-2-1-3, 10-3-1-6, 10-5-1-10 and 20-2-2-3. The results are displayed in Table 4.13. In the table, columns 2-6 show the average CPU time (in seconds) required by CPLEX to solve all the instances using the formulation stated in the respective header and when applying the change given in column 1. Recall that “Base case” shows the result with the original values of the parameters, “TC” indicates that we increased the travel cost of each edge of the graph by 10, “SR” shows the results when we set the same revenue for all the ages, “SH” shows the results when we set the same value for the holding costs of all facilities and ages, “ZH” presents the results when we set the holding costs as zero for all facilities and ages, “DA” shows the results when we double the maximum age of the product, “RSC” displays the results when we reduce by 20% the storage capacity of the customers, and “RVC” shows the results when we reduce the vehicle capacities using a predefined filling coefficient.

All but one instance could be solved to optimality in less than two hours using all the formulations (namely, one instance of set 10-5-1-10 when solved by formulation TP-I-nk and applying change SR). In this case, we computed the average time by setting a CPU time of 7200 seconds for this instance (which is indicated by the “*” mark in the table). It is worth mentioning that we could not solve this instance with the applied change and formulation TP-I-nk even within a considerably larger running time limit (14 hours).

In the table, we observe that when we increase the routing cost (TC) and when we set the same revenue for all the ages (SR), the total time required to solve the instances increased for all the formulations (compared to the base case). The increase is especially noticeable for the change SR. This may be due to the observation that now changes on the integer decision variables (x and y) have a reduced effect on the objective function (total profit), which directly affects the performance of the B&C algorithm. On the other hand, it is possible to observe that the time required by the solver decreases for scenarios SH, ZH, and RSC, compared to the base case. For scenarios SH and ZH, the time reduction can be explained because the inventory part of the problem does not have any impact on the objective function value anymore, remaining only routing and consumption decisions as key to maximizing it. Thus, the problem reduces to maximizing the amount of fresh product that is sent to the customers, as long as it is profitable by the revenue values. For scenario RSC the time reduction may be due to the tighter bounds on the delivery and inventory variables, which depend on the storage capacity parameter.

For scenario DA, the running time remains relatively stable for formulations AB, TP-I and TP-II, while for the formulations without a vehicle index (TP-I-nk and TP-II-nk), which have SECs that include the delivery variables (q), the total running time increased significantly. The latter observation may be explained by fact that it is now harder to find feasible solutions due to the considerable larger number of delivery variables (which depend on the maximum age) together with the SECs that include the delivery variables. Finally, for scenario RVC, the total time increases considerably for the formulations without a vehicle index, possibly given that more SECs have to be separated for these formulations as the capacity is tighter now and they depend on the vehicle capacity parameter. For formulation AB the increase in the CPU times can be explained given that now it is harder to find a feasible solution for the solver, given the capacity tightness. For formulations TP-I and TP-II the total remains relatively stable. It is worth mentioning that these conclusions must be taken cautiously given that some variability on the results may appear as a result of (slight) differences on the cores of the computer grid used to run these experiments.

	Formulation				
	AB	TP-I	TP-I-nk	TP-II	TP-II-nk
Base case	11.82	6.37	10.44	7.21	6.85
TC	43.17	28.71	55.57	29.02	284.88
SR	111.46	127.90	645.77*	85.84	496.27
SH	7.91	2.95	1.01	3.46	1.07
ZH	5.43	2.66	1.47	2.46	1.32
DA	9.43	7.08	34.41	6.41	39.97
RSC	3.59	2.96	3.08	3.56	4.02
RVC	23.98	6.82	45.95	6.93	38.30

Table 4.13: CPU times when the parameters of the problem instances are changed

4.6 Final remarks

In this chapter, we addressed an inventory routing problem in which goods are perishable. We present four new mathematical formulations for the problem, two with a vehicle index and two without a vehicle index, and present branch-and-cut algorithms to solve them. We also developed a hybrid solution method for the problem by combining an iterated local search metaheuristic with two mathematical programming components. Additionally, we introduced new instances for the problem. The results of the computational experiments show that the formulations without a vehicle index provide a considerably larger number of feasible solutions within two hours when compared to the other formulations, in addition to a significant speed-up for instances solved to optimality within the time limit by all the formulations. Furthermore, our hybrid heuristic solution method was able to provide high-quality solutions within relatively short running times on small- and medium-sized problem instances. When applied to larger instances, the method provides good feasible solutions within reasonable running times.

Chapter 5

The stochastic IRP under supply and demand uncertainty

In this chapter, we address the stochastic inventory routing problem under the consideration that both the product supply and the customer demands are uncertain. We propose a two-stage stochastic programming formulation, where routing decisions are made in the first stage, while delivery quantities, inventory levels and specific recourse actions are determined in the second stage. We consider different recourse mechanisms such as lost sales and backlogging as well as an additional source for the product in a capacity reservation contract setting. The objective is to minimize the first-stage cost plus the total expected inventory and recourse cost incurred in the second stage. We also propose a heuristic solution method which is based on the progressive hedging algorithm. We provide managerial insights resulting from extensive computational experiments using instances based on a benchmark test set from the literature. In particular, we study the response mechanisms of the optimal solutions under different uncertainty levels of the random variables and different cost configurations. The results of the heuristic method show that it provides high-quality solutions within reasonable running times for instances with a large number of scenarios.

⋆ An article based on the contents of this chapter is published as:

Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., and Morabito, R. (2020). Inventory routing under stochastic supply and demand, Technical Report, *Les Cahiers du GERAD*, G-2020-14, HEC Montréal, Canada.

5.1 Introduction

In this chapter, we address the stochastic inventory routing problem (SIRP) in the context where both the product supply and the customer demands are uncertain. We consider the basic variant of the inventory routing problem (IRP) in a one-to-many setting, in which a single central supplier has to serve the demand of multiple customers in every period of a specified time horizon. In each period, the supplier can use a fleet of vehicles to deliver the product to the customers, while minimizing the total cost of the system. Whereas demand uncertainty has been studied before in the context of the IRP, to the best of our knowledge supply uncertainty has not yet been addressed. We use a two-stage decision framework in which the routing decisions are made in the first stage while the delivery quantities, inventory levels, and specific recourse actions are determined in the second stage.

Uncertainty plays a crucial role in supply chain management given that critical input data which are required for effective planning often are not known with certainty when the plan is made, which directly impacts the quality of the decisions. Since using inaccurate information can lead to poor performance in practice, it becomes relevant to take uncertainty into account in the decision process. In the IRP context, demand (or downstream) uncertainty appears naturally because of seasonality, changes in customer preferences, and inaccurate forecasts, among others. On the other hand, supply (or upstream) uncertainty can arise from delays or shortages from the supply source or from disruptions at the supplier's production plant. In particular, supply uncertainty may play a critical role in this context given that we consider the problem variant with a single supplier. This centralization of the service implies that even relatively small supply disruptions at the supplier will affect the service to the customers. This is often disregarded in stochastic IRP studies.

Several variants of the SIRP have been studied over the past decades. Most of them (including ours) differ by the specific features that they consider, such as the random variables that are modeled, and the type of planning horizon considered, among others. We review them to put our contribution into perspective. However, given the multiple extensions of the IRP that can be explored in practice, we only review SIRPs considering demand uncertainty in a finite planning horizon for road-based transportation applications. We are not aware of any work addressing the basic variant of the IRP with supply uncertainty. For extensive reviews on the IRP, including stochastic components, we refer the reader to Andersson et al. (2010) and Coelho et al. (2014b). For studies exploring infinite horizon problems we refer the reader to the works of Jaillet et al. (2002), Kleywegt et al. (2004) and Hvattum et al. (2009). Also, it is worth mentioning that supply uncertainty has been explored in other contexts such as humanitarian logistics (Moreno et al., 2018), supply chain design and planning (Zeballos et al., 2014) and supplier selection (Burke et al., 2009).

Federgruen and Zipkin (1984) were the first to consider demand uncertainty in the IRP context. They addressed a single-period SIRP, including inventory holding, shortage and transportation costs. Federgruen et al. (1986) extended this work by considering a perishable product and incorporating spoilage costs. The authors studied two different transportation alternatives:

direct deliveries performed individually to each customer, and multi-customer routes carried out by a fleet of vehicles. Huang and Lin (2010) studied a single-period multi-product SIRP with uncertain demands and stockouts. The authors assumed that the demands only become known upon the arrival of the vehicle at the customer locations and included a recourse mechanism consisting of a return trip to the depot when stockouts occur. The objective function consists of minimizing the total cost given by the sum of planned routes, the recourse costs and expected stockout costs. The authors presented a modified ant colony optimization metaheuristic. Yu et al. (2012) addressed an IRP with split deliveries and stochastic demands. The authors included service level constraints imposing (with a given probability) both demand fulfillment and maximum storage capacity usage at the customer facilities. The authors proposed a hybrid solution approach that combines the simplification of an approximate model of the problem as well as repair and local search operators.

Bertazzi et al. (2013) addressed an IRP with stochastic demands and stockouts under an order-up-to-level replenishment policy, i.e., whenever a customer is visited, the quantity delivered is such that its inventory level reaches the maximum storage capacity. The objective function minimizes the expected total cost given by the sum of the expected inventory, out-of-stock penalties and routing costs. The authors developed a dynamic programming formulation of the problem and a rollout algorithm. Bertazzi et al. (2015) addressed a similar problem but considering that the deliveries are performed using transportation procurement. Coelho et al. (2014a) addressed the IRP under the assumption of dynamic and stochastic demands. The authors proposed different heuristic policies for the problem. A single vehicle is used and transshipments between customers are allowed. The objective function consists of minimizing the sum of inventory, shortage, routing and transshipment costs. Nolz et al. (2014) addressed an IRP appearing in a medical waste collection application where demands are stochastic. In the problem, a single vehicle is used to pick up medical waste boxes from pharmacies. The authors proposed a two-stage stochastic programming formulation for the problem and a heuristic method based on an adaptive large neighborhood search algorithm. The objective function minimizes the sum of fixed and variable routing costs plus the expected second-stage cost given by the sum of excess inventory costs and penalty costs imposed for picking up less than a given threshold when visiting a pharmacy.

Gruler et al. (2018) addressed a single-period IRP with stochastic demands and stockouts. The objective function consists of minimizing the sum of expected inventory and routing costs. The authors presented a simheuristic, based on the combination of a variable neighborhood search metaheuristic with simulation. Nikzad et al. (2019) addressed a stochastic IRP appearing in medical drug distribution with uncertain demands. The objective function minimizes the sum of inventory, transportation and stockout costs. The authors presented a two-stage stochastic programming formulation and two chance-constrained stochastic formulations. A matheuristic solution algorithm is proposed. Markov et al. (2018) presented a unified framework for various classes of rich routing problems with stochastic demands, including, among others, different classes of IRPs (health care, waste collection and maritime IRP). The framework includes real-world demand forecasting techniques to provide the model with the expected demands. The authors also explicitly modeled undesirable events as well as recourse actions. Markov et al. (2020) addressed an IRP with stochastic demands appearing in a recyclable waste collection application. The authors developed an adaptive large neighborhood search algorithm and used a realistic demand forecasting model to estimate the expected demands and the uncertainty levels, as in Markov et al. (2018).

IRPs for perishable products with stochastic demands and fixed deterministic shelf-lives were studied by Soysal et al. (2015), Soysal et al. (2018) and Crama et al. (2018). In the problem addressed by Soysal et al. (2015) the objective function consists of minimizing the total cost, given by the sum of routing, inventory and waste costs. The problem allows unmet demands to be backlogged and multiple visits to the customers in each time period. The authors proposed several chance-constrained models for the problem. Soysal et al. (2018) extended this study by considering multiple perishable products and collaboration among different suppliers. They presented a chance-constrained formulation and a deterministic approximate formulation of the chance-constrained program. Crama et al. (2018) studied a problem including a maximum duration for the vehicle routes and target service levels. In their problem, the objective function consists of maximizing the expected profit given by the total sales revenue minus the acquisition, distribution, and other miscellaneous costs. The authors proposed several solution methods for the problem, namely an expected value method, a deliver-up-to-level method, a decomposition method relying on a stochastic dynamic programming model, and a decomposition-integration method.

The contributions of this chapter are threefold. First, we introduce a two-stage stochastic programming formulation for the SIRP under uncertain supply of the product and uncertain customer demands. This formulation can be adapted to consider different recourse mechanisms, such as lost sales, backlogging and an additional supply source in a capacity reservation contract setting. We study for the first time supply uncertainty in the context of the basic variant of the IRP as well as a capacity reservation contract setting in the IRP context. As a second contribution, we present a heuristic solution method based on the progressive hedging algorithm. This method provides high-quality solutions within reasonable running times for instances with a large number of scenarios. Our final contribution consists of providing managerial insights from experiments using instances based on a benchmark test set from the literature. In particular, we study the response mechanisms of the optimal solutions for different levels of uncertainty and cost configurations. Furthermore, we observe that supply and demand uncertainty have different effects on the value of taking the uncertainty into account. We also study the effect of incorporating a service level.

The remaining sections of this chapter are organized as follows. In Section 5.2, we describe the problem and introduce the mathematical notation. In Section 5.3 we present the formulations for the different cases of the problem, and in Section 5.4 we present our heuristic method for the same cases. In Section 5.5 we describe the computational experiments that we performed and discuss the results. Finally, Section 5.6 highlights the final remarks of the chapter.

5.2 Problem description

The two-stage SIRP consists of a supplier, whose depot is denoted by node 0, who has to serve the demand of N customers in each one of the T periods of a specified time horizon. In this problem, the customers are represented by the set $\mathcal{C} = \{1, \ldots, N\}$ and the time horizon by $\mathcal{T} = \{1, \ldots, T\}$. To serve the customer demands, the supplier can use up to K vehicles, each one having a capacity Q. All the vehicles are based at the depot and are represented by the set $\mathcal{K} = \{1, \ldots, K\}$. The vehicle routes take place in a distribution network represented by the set of arcs $\mathcal{A} = \{(i, j) \colon i, j \in \mathcal{N}, i \neq j\}$, where $\mathcal{N} = \{0\} \cup \mathcal{C}$ is the set of all the facilities of the system (supplier's depot and its customers). Every route starts from and must return to the depot. Also, a routing cost c_{ij} is incurred every time a vehicle traverses arc $(i, j) \in \mathcal{A}$. An inventory holding cost h_i^t has to be paid for every unit of product in stock at the end of each period $t \in \mathcal{T}$ in each facility $i \in \mathcal{N}$. In addition, there is an initial amount I_i^0 available in every facility $i \in \mathcal{N}$ at the beginning of the time horizon. Finally, the stock at hand at each customer $i \in \mathcal{C}$ is restricted by the maximum storage limit C_i.

The supply and demands are random variables with known discrete probability distributions (assuming independence for all facilities and periods). Let $\mathcal{S}$ denote the finite set of all the possible scenarios (supply and demand realizations), and let ρ_s be the probability of occurrence of scenario $s \in \mathcal{S}$, with $\rho_s > 0$, $\forall s \in \mathcal{S}$ and $\sum_{s \in \mathcal{S}} \rho_s = 1$. Let d_{is}^t be the demand of customer $i \in \mathcal{C}$ in time period $t \in \mathcal{T}$ under scenario $s \in \mathcal{S}$ and let r_s^t be the amount of product the supplier receives in time period $t \in \mathcal{T}$ under scenario $s \in \mathcal{S}$.

In the SIRP, any quantity can be delivered to the customers as long as the maximum holding capacity is not exceeded. In addition, we work under the following assumptions: the storage capacity of the supplier is large enough to store all the received amounts at the depot; the demand of a given time period can be satisfied with a delivery performed in the same period; and the amount the supplier receives in each period can be used to perform deliveries in that same period. We also assume that the supply and demand realizations are known before all the vehicles depart from the supplier depot. Therefore, the SIRP consists of determining, in the first stage, the vehicle routes that will be performed in each time period and, after the realization of the supply and demand scenario (second stage), the delivery quantities and the required recourse decisions (if any) such that the total cost is minimized. This total cost is given by the first-stage cost (vehicle routing cost) plus the expected cost of the second-stage decisions. Figure 5.1 shows the timing of the events that we assume in this chapter.

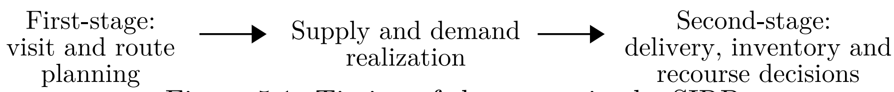


Figure 5.1: Timing of the events in the SIRP

Adulyasak et al. (2015a) pointed out that this type of setting follows real-world practice, in which some decisions are planned beforehand using information about possible values of the input data (e.g., product availability and customer demands) and these plans remain fixed in the execution phase. This is done with the aim of designing plans that are less sensitive to data uncertainty (Vladimirou and Zenios, 1997) and also to maintain consistency in the planned activities and to avoid large disruptions in the initial plan. Particularly, consistency represents an important issue in distribution planning activities since typically the planned visits have to be informed in advance to the customers and the required resources need to be prepared (Kovacs et al., 2014; Coelho et al., 2012). It is worth mentioning that a similar timing of events has been used in other studies (Nikzad et al., 2019; Adulyasak et al., 2015a; Nolz et al., 2014).

5.3 Two-stage stochastic programming formulations

In this section, we describe the mathematical formulations that we introduce for the SIRP. First, in Sections 5.3.1 and 5.3.2 we present formulations for the SIRP with lost sales and backlogging as recourse decisions, respectively. Then, in Section 5.3.3, we describe a mathematical formulation for the SIRP with a capacity reservation contract (CRC), which can be used as an additional recourse mechanism in the second stage, but also requires an additional first-stage decision.

5.3.1 Lost sales formulation

To model the SIRP with lost sales, we introduce the parameter a_i, which is the penalty incurred by the supplier for every unit of unmet demand at customer $i \in \mathcal{C}$ in each time period. This penalty can be interpreted as the opportunity cost for the stockouts or as the outsourcing cost paid to a third-party responsible for delivering the product to the customers. Also, we introduce the following decision variables:

$x_{ij}^{kt} \in \{0, 1\}$: 1 if vehicle k traverses arc (i, j) in time period t, 0 otherwise;
$y_i^{kt} \in \{0, 1\}$: 1 if facility i is visited by vehicle k in period t, 0 otherwise;
$I_{is}^t \geq 0$: inventory at facility i at the end of time period t under scenario s;
$q_{is}^{kt} \geq 0$: quantity delivered to customer i by vehicle k in period t under scenario s;
$u_{is}^t \geq 0$: unmet demand at customer i in time period t under scenario s.

Given these decision variables, the formulation can be stated as follows:

$$\min \quad \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} c_{ij} x_{ij}^{kt} + \sum_{s\in\mathcal{S}} \rho_s \left(\sum_{i\in\mathcal{N}} \sum_{t\in\mathcal{T}} h_i^t I_{is}^t + \sum_{i\in\mathcal{C}} \sum_{t\in\mathcal{T}} a_i u_{is}^t \right) \tag{5.1}$$

$$\text{s.t.} \quad I_{0s}^t = I_{0s}^{t-1} + r_s^t - \sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}} q_{is}^{kt} \qquad t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.2}$$

$$I_{is}^t = I_{is}^{t-1} + \sum_{k\in\mathcal{K}} q_{is}^{kt} + u_{is}^t - d_{is}^t \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.3}$$

$$I_{is}^{t-1} + \sum_{k\in\mathcal{K}} q_{is}^{kt} \leq C_i \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.4}$$

$$q_{is}^{kt} \leq \min\{Q, C_i\} y_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.5}$$

$$\sum_{i\in\mathcal{C}} q_{is}^{kt} \leq Q y_0^{kt} \qquad k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.6}$$

$$\sum_{j \in \mathcal{N}: j \neq i} x_{ji}^{kt} = y_i^{kt} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, \quad (5.7)$$

$$\sum_{j \in \mathcal{N}: j \neq i} x_{ij}^{kt} = y_i^{kt} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, \quad (5.8)$$

$$\sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}: j \neq i} x_{ij}^{kt} \leq \sum_{i \in \mathcal{B}} y_i^{kt} - y_l^{kt} \qquad \forall \mathcal{B} \subseteq \mathcal{C}, |\mathcal{B}| \geq 2, k \in \mathcal{K}, t \in \mathcal{T}, l \in \mathcal{B}, \quad (5.9)$$

$$\sum_{k \in \mathcal{K}} y_i^{kt} \leq 1 \qquad i \in \mathcal{C}, t \in \mathcal{T}, \quad (5.10)$$

$$I_{is}^t \geq 0 \qquad i \in \mathcal{N}, t \in \mathcal{T}, s \in \mathcal{S}, \quad (5.11)$$

$$q_{is}^{kt} \geq 0 \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \quad (5.12)$$

$$u_{is}^t \geq 0 \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}, \quad (5.13)$$

$$y_i^{kt} \in \{0, 1\} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, \quad (5.14)$$

$$x_{ij}^{kt} \in \{0, 1\} \qquad (i, j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}. \quad (5.15)$$

The objective function (5.1) consists of minimizing the total cost, given by the sum of transportation costs (first-stage) and inventory holding and lost sales costs (second-stage). Constraints (5.2) and (5.3) balance the inventory at the supplier and the customers, respectively, for every scenario. Constraints (5.4) impose the customers' storage capacity. Constraints (5.5) link delivery and visit variables. Constraints (5.6) enforce the capacity of the vehicles in every scenario. Constraints (5.7) and (5.8) are the vehicle flow conservation while constraints (5.9) are the subtour elimination constraints (SECs). Constraints (5.10) limit the number of visits to each customer every time period. Finally, the domain of the decision variables is defined in constraints (5.11)-(5.15).

5.3.2 Backlogging formulation

Instead of assuming that all the unmet demand is immediately lost, we can model a situation in which the unmet demand can be delivered in later periods with an associated penalty cost, i.e., we can use backlogging as the recourse action in the second stage. We introduce the variable B_{is}^t which is the amount backlogged at customer $i \in \mathcal{C}$ in time period $t \in \mathcal{T}$ under scenario $s \in \mathcal{S}$. Additionally, let a_i be now the backlogging cost at customer $i \in \mathcal{C}$ in each time period. Then, the two-stage stochastic programming formulation for the SIRP with backlogging can be stated as follows:

$$\min \quad \sum_{(i,j) \in \mathcal{A}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} c_{ij} x_{ij}^{kt} + \sum_{s \in \mathcal{S}} \rho_s \left(\sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} h_i^t I_{is}^t + \sum_{i \in \mathcal{C}} \sum_{t \in \mathcal{T}} a_i B_{is}^t \right) \quad (5.16)$$

$$\text{s.t.} \quad I_{is}^t - B_{is}^t = I_{is}^{t-1} - B_{is}^{t-1} + \sum_{k \in \mathcal{K}} q_{is}^{kt} - d_{is}^t \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}, \quad (5.17)$$

$$B_{is}^t \geq 0 \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}, \quad (5.18)$$

$$(5.2), (5.4)\text{-}(5.12) \text{ and } (5.14)\text{-}(5.15).$$

The objective function (5.16) consists of minimizing the total cost, given by the sum of routing costs and expected inventory holding and backlogging costs. Constraints (5.17) define the inventory conservation at the customers, now considering the backlogging variable, and constraints (5.18) define the domain of the new decision variable. Notice that we do not force the backlogging to be null in the last period, which can be viewed as a lost sales allowance at the end of the planning horizon, or a backlogged amount that will be carried over to the next planning horizon.

5.3.3 Capacity reservation contract formulation

In this section, we model the SIRP with a CRC as an additional recourse mechanism. In this setting, the supplier can make a contract with an external provider to reserve a certain amount of manufacturing capacity upfront, such that the external provider is able to produce any amount within the limits of the reserved capacity in each period (Serel et al., 2001). The external provider delivers the extra amount directly to the supplier, who subsequently transports this to the customers. When a CRC is established, the purchasing cost for the supplier is typically lower than the expected cost on the spot market (in our case, lost sales or backlogging costs). However, the supplier has to pay a certain amount upfront for this capacity reservation, irrespective of whether this capacity will be later used or not. CRCs allow the supplier to reduce purchasing costs and shortage risks, while the resource utilization of the external provider is increased. This type of business-to-business arrangement is especially useful in uncertain environments and can be found in many industries such as commodity chemicals or semiconductor manufacturing (Kleindorfer and Wu, 2003; Serel, 2007).

In our problem, if the supplier contracts this additional capacity, it incurs a fixed reservation cost in the first stage and a variable cost for each unit of the reserved capacity that is actually used in the second stage. The contracted supplementary capacity is a multiple of a base capacity Δ offered by the external source. The extra amount that can be made available at the supplier facility at the beginning of any time period of the planning horizon is a supplementary recourse in the second stage, together with lost sales or backlogging, as in Sections 5.3.1 and 5.3.2, respectively.

To model this problem, we consider the following additional notation. Let f be a fixed reservation cost incurred for each Δ units of external capacity contracted and p be the procurement cost for each extra unit actually acquired by the supplier. We also introduce the following decision variables:

$z \in \mathbb{Z}^+$: number of times the base capacity Δ is contracted by the supplier in the first stage;

$w_s^t \geq 0$: extra amount made available at the supplier at the beginning of period t under scenario s.

To illustrate the relation between the capacity reservation variable (z) and the product availability and reservation cost in the CRC setting, Figure 5.2 shows a numerical example. Assume a base capacity of 10 units and a fixed reservation cost equal to 5 (i.e., $\Delta = 10$ and $f = 5$). The chart shows, for different values of the reservation variable (z) in the horizontal axis, the first-stage cost incurred by the contract made, given by fz; and the maximum extra amount that the supplier can use every time period, computed as $z\Delta$.

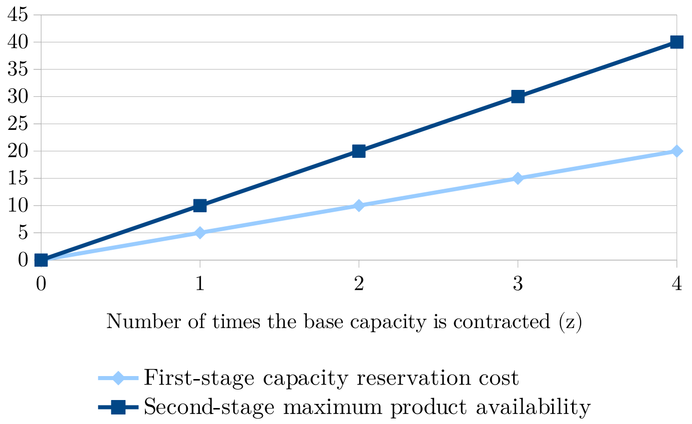


Figure 5.2: Product availability and reservation cost for different number of contracts

Given this notation, the two-stage stochastic programming formulation for the SIRP with CRC and lost sales can be stated as follows:

$$\min \quad \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} c_{ij} x_{ij}^{kt} + fz + \sum_{s\in\mathcal{S}} \rho_s \left(\sum_{i\in\mathcal{N}} \sum_{t\in\mathcal{T}} h_i^t I_{is}^t + \sum_{i\in\mathcal{C}} \sum_{t\in\mathcal{T}} a_i u_{is}^t + \sum_{t\in\mathcal{T}} p w_s^t \right) \tag{5.19}$$

$$\text{s.t.} \quad I_{0s}^t = I_{0s}^{t-1} + r_s^t + w_s^t - \sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}} q_{is}^{kt} \qquad t\in\mathcal{T}, s\in\mathcal{S}, \tag{5.20}$$

$$w_s^t \leq z\Delta \qquad t\in\mathcal{T}, s\in\mathcal{S}, \tag{5.21}$$

$$w_s^t \geq 0 \qquad t\in\mathcal{T}, s\in\mathcal{S}, \tag{5.22}$$

$$z \in \mathbb{Z}, \tag{5.23}$$

(5.3)-(5.15).

The objective function (5.19) consists of minimizing the total cost, given by the sum of routing and capacity reservation costs (first stage) and expected second-stage cost, given by the inventory holding and lost sales costs plus the total procurement of the extra amounts actually acquired by the supplier. Constraints (5.20) define the inventory conservation at the supplier, in which there are now two sources of the product: the regular source (parameter r) and the extra source (variable w). Constraints (5.21) allow the usage of the additional capacity in each time period up to the level actually reserved in advance (first stage) by the supplier. Constraints (5.22) and (5.23) define the domain of the usage and reservation decision variables, respectively.

Note that this model also uses the lost sales variables (u), as used in Section 5.3.1. The variant for this formulation considering backlogging instead of lost sales can be obtained by modifying the inventory conservation constraints of the customers to include the backlogging variable (B), as in Section 5.3.2. Also, it is worth mentioning that the model can be extended to a more general approach with several (different) modular capacities, e.g., L alternatives offered in base capacities $\Delta_1, \Delta_2, \ldots, \Delta_L$, of which there are $U_1, U_2, \ldots, U_L$ units available for reservation, respectively. Integer variables $z_1, z_2, \ldots, z_L$, bounded by their respective availability ($z_\ell \leq U_\ell,\ \ell = 1, \ldots, L$), can be defined to indicate the usage of the different modular alternatives. Finally, the extra amount made available in each time period (w_s^t) would be bounded by the sum of the reserved capacities, i.e., $w_s^t \leq \sum_{\ell=1}^{L} z_\ell \Delta_\ell$.

5.3.4 Remarks on the computational implementation

We provide a few remarks regarding the computational implementation of the proposed formulations. First of all, note that we used a complete directed graph to represent the problem, with an arc set $\mathcal{A} = \{(i,j)\colon i,j \in \mathcal{N}, i \neq j\}$. However, when the travel costs are assumed to be symmetric, i.e., $c_{ij} = c_{ji},\ \forall (i,j) \in \mathcal{A}$, the arc set can be replaced with the following edge set $\mathcal{E} = \{(i,j)\colon i,j \in \mathcal{N}, i < j\}$. This allows us to model the problem using a considerably smaller number of vehicle flow variables x, which positively impacts the performance of the solver. The constraints involving these variables must be modified accordingly.

Regarding the separation of the SECs (5.9), which are used in the three formulations, we use an exact procedure that relies on several minimum cut problems, which are solved using the Concorde solver (Applegate et al., 2018). For this, consider the following notation. Let $\bar{y}_i^{kt}$ and $\bar{x}_{ij}^{kt}$ represent, for a given solution (fractional or integer) found during the branch-and-cut (B&C) process, the values for the variables y_i^{kt} and x_{ij}^{kt}, respectively. The algorithm then builds a graph for every pair (k,t) with $\bar{y}_0^{kt} > 0$. The vertex set of each graph consists of all the nodes $i \in \mathcal{C}$ of the solution for which $\bar{y}_i^{kt}$ takes a positive value. The weight of the edges of each graph is set to $\bar{x}_{ij}^{kt}$, for every pair of vertices of the corresponding graph. Then, for each customer vertex of the constructed graph, the separation procedure solves a minimum $s-t$ cut problem, where the source vertex is set as the supplier node ($s = 0$) and the sink vertex is set as the customer node ($t = i$). If the minimum cut capacity is less than $2\bar{y}_i^{kt}$ then a violated SEC has been found (Adulyasak et al., 2014a; Alvarez et al., 2020). The set $\mathcal{B}$ of the respective equation contains the nodes of the minimum cut and we add constraints (5.9) with $\ell = \arg\max_{i \in \mathcal{B}}\{\bar{y}_i^{kt}\}$, for every vehicle and time period. This separation procedure is applied only at the root node and to integer solutions, in order to work with a reduced number of cuts.

We also explored symmetry breaking constraints (SBCs) in our implementation. This is an important issue given that there can be a large number of symmetric solutions in each time period due to the homogeneous nature of the vehicle fleet, which negatively impacts the performance of the B&C algorithm. We explored the SBCs used by Adulyasak et al. (2014a) and Coelho and Laporte (2013b). In particular, in our implementation we used the following SBCs:

$$y_0^{kt} \leq y_0^{k-1,t} \qquad k \in \mathcal{K}\backslash\{1\}, t \in \mathcal{T}, \tag{5.24}$$

$$\sum_{i=1}^{j} 2^{(j-i)} y_i^{kt} \leq \sum_{i=1}^{j} 2^{(j-i)} y_i^{k-1,t} \qquad j \in \mathcal{C}, k \in \mathcal{K}\backslash\{1\}, t \in \mathcal{T}. \tag{5.25}$$

Constraints (5.24) allow the use of vehicle k only if vehicle $k-1$ is used in the same time period. Constraints (5.25) belong to the lexicographic ordering constraints family (Jans, 2009). These constraints assign a unique number to each possible subset of customers on a route and order the vehicles according to this number. This combination of SBCs was chosen after several preliminary experiments using a subset of the instances with two vehicles. A similar combination of SBCs was also used by Adulyasak et al. (2014a).

Notice also that we can model the problem using a formulation without a vehicle index, since the fleet is considered to be homogeneous in terms of capacity and travel times and costs. However, in that case we would need either MTZ-like SECs or capacity cuts as both SECs and vehicle capacity constraints. These capacity cuts would have the form shown in (5.26).

$$Q\sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B}:j\neq i} x_{ij}^t \leq Q\sum_{i\in\mathcal{B}} y_i^t - \sum_{i\in\mathcal{B}} q_{is}^t \quad \forall\mathcal{B}\subseteq\mathcal{C}, |\mathcal{B}|\geq 2, t\in\mathcal{T}, s\in\mathcal{S}. \tag{5.26}$$

The former option would yield a considerably weaker formulation. On the other hand, the latter option results in capacity cuts including the delivery quantity variables, which are second-stage variables. Thus, it would be necessary to separate the capacity cuts also for each scenario, which becomes prohibitive when the number of scenarios is large.

5.4 A progressive hedging-based heuristic for the SIRP

This section presents the heuristic algorithm that we propose to solve the SIRP, which is based on the progressive hedging (PH) algorithm (Rockafellar and Wets, 1991). In their work, Rockafellar and Wets proposed a scenario-based decomposition method for stochastic programs based on an augmented Lagrangean strategy. The method solves a series of subproblems resulting from the scenario decomposition and guides the search to find a solution in which the aggregation of the subproblem solutions is non-anticipative (i.e., the first-stage solution is not scenario-dependent) and optimal. The authors proved that their method converges to a global optimum in the convex case and showed that if it converges in the nonconvex case when the subproblems are solved to local optimality then the resulting solution is a local optimum. Several PH-based heuristics have been proposed in the literature for stochastic problems with integer variables, e.g., Løkketangen and Woodruff (1996) for mixed integer $(0, 1)$ multi-stage stochastic programs, Haugen et al. (2001) for stochastic lot-sizing problems, Crainic et al. (2011) for a stochastic network design problem and Lamghari and Dimitrakopoulos (2016) for open-pit mine production scheduling under uncertainty.

In our approach, when we apply the scenario decomposition to the original stochastic problem (Section 5.4.1) it results in a series of subproblems that take the form of a deterministic IRP with visiting costs for each scenario s (which we will refer to as IRP(s)). These problems are solved using an iterated local search (ILS)-based hybrid method (Section 5.4.4). In each outer iteration of the algorithm, the cost parameters of each scenario are adjusted to reflect the differences between the scenario solution and a reference solution (Section 5.4.3). These adjustments are made with the aim of reaching a consensus on the first-stage solutions over all the scenarios and thus to a feasible solution for the complete stochastic problem. The reference solution is constructed from the solution of all the scenarios in the previous iteration (Section 5.4.2). All the components of the overall approach are described in the upcoming sections. It is worth mentioning that in this section we describe the application of this method for the SIRP with lost sales, but the procedure can be applied analogously for both the backlogging case and the CRC case.

5.4.1 Scenario decomposition for the SIRP

One can see that the formulation presented in Section 5.3.1 has a block-angular structure (each block representing a deterministic IRP with lost sales for each scenario $s \in \mathcal{S}$) with constraints (5.5) and (5.6) linking the first- and second-stage variables. These linking constraints forbid the delivery quantities of every scenario to take positive values when the respective first-stage variables are zero. Thus, we can reformulate the problem after creating a copy of the first-stage variables for each scenario $s \in \mathcal{S}$ (x_{ijs}^{kt} and y_{is}^{kt}), as follows:

$$\min \quad \sum_{s \in \mathcal{S}} \rho_s \left(\sum_{(i,j) \in \mathcal{A}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} c_{ij} x_{ijs}^{kt} + \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} h_i^t I_{is}^t + \sum_{i \in \mathcal{C}} \sum_{t \in \mathcal{T}} a_i u_{is}^t \right) \tag{5.27}$$

$$\text{s.t.} \quad q_{is}^{kt} \leq \min\{Q, C_i\} y_{is}^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.28}$$

$$\sum_{i \in \mathcal{C}} q_{is}^{kt} \leq Q y_{0s}^{kt} \qquad k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.29}$$

$$\sum_{j \in \mathcal{N}: j \neq i} x_{jis}^{kt} = y_{is}^{kt} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.30}$$

$$\sum_{j \in \mathcal{N}: j \neq i} x_{ijs}^{kt} = y_{is}^{kt} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.31}$$

$$\sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}: j \neq i} x_{ijs}^{kt} \leq \sum_{i \in \mathcal{B}} y_{is}^{kt} - y_{\ell s}^{kt} \qquad \forall \mathcal{B} \subseteq \mathcal{C}, |\mathcal{B}| \geq 2, k \in \mathcal{K}, t \in \mathcal{T}, \ell \in \mathcal{B}, s \in \mathcal{S}, \tag{5.32}$$

$$\sum_{k \in \mathcal{K}} y_{is}^{kt} \leq 1 \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.33}$$

$$y_{is}^{kt} = \hat{y}_i^{kt} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.34}$$

$$x_{ijs}^{kt} = \hat{x}_{ij}^{kt} \qquad (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.35}$$

$$y_{is}^{kt} \in \{0,1\} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.36}$$

$$x_{ijs}^{kt} \in \{0,1\} \qquad (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.37}$$

$$\hat{y}_i^{kt} \in \{0,1\} \qquad i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{5.38}$$

$$\hat{x}_{ij}^{kt} \in \{0,1\} \qquad (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{5.39}$$

(5.2)-(5.4) and (5.11)-(5.13).

Constraints (5.34) and (5.35) ensure that the first-stage solutions will be the same for all the scenarios. These constraints are imposed to guarantee that a single "implementable" solution will be obtained (Rockafellar and Wets, 1991), i.e., a single set of vehicle routes (and their respective visit decisions) for each time period over all the scenarios, instead of scenario-tailored first-stage solutions. The variables $\hat{y}_i^{kt}$ and $\hat{x}_{ij}^{kt}$ are referred to as the "overall" first-stage variables (Crainic et al., 2011).

Following the separation procedure of the PH algorithm, constraints (5.34) and (5.35) are relaxed using an augmented Lagrangean method, alternatively referred to as multiplier method (Luenberger and Ye, 2016), which results in the following objective function for the formulation:

$$\min \sum_{s\in\mathcal{S}} \rho_s \Bigg(\sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} c_{ij} x_{ijs}^{kt} + \sum_{i\in\mathcal{N}} \sum_{t\in\mathcal{T}} h_i^t I_{is}^t + \sum_{i\in\mathcal{C}} \sum_{t\in\mathcal{T}} a_i u_{is}^t + \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \lambda_{is}^{kt} (y_{is}^{kt} - \hat{y}_i^{kt}) + \frac{1}{2} \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \delta (y_{is}^{kt} - \hat{y}_i^{kt})^2 + \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \mu_{ijs}^{kt} (x_{ijs}^{kt} - \hat{x}_{ij}^{kt}) + \frac{1}{2} \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \delta (x_{ijs}^{kt} - \hat{x}_{ij}^{kt})^2 \Bigg), \tag{5.40}$$

with unrestricted multipliers λ_{is}^{kt} and μ_{ijs}^{kt} for the relaxed constraints (5.34) and (5.35), respectively, and a penalty term δ. They penalize the difference of the values of the visit and routing decisions between the scenario solution (y_{is}^{kt} and x_{ijs}^{kt}) and the "overall" first-stage variables ($\hat{y}_i^{kt}$ and $\hat{x}_{ij}^{kt}$). Then, given that the variables x_{ijs}^{kt} and y_{is}^{kt} are binary, the function can be reduced as follows:

$$\min \sum_{s\in\mathcal{S}} \rho_s \Bigg(\sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} (c_{ij} + \mu_{ijs}^{kt} + \frac{1}{2}\delta - \delta \hat{x}_{ij}^{kt}) x_{ijs}^{kt} + \sum_{i\in\mathcal{N}} \sum_{t\in\mathcal{T}} h_i^t I_{is}^t + \sum_{i\in\mathcal{C}} \sum_{t\in\mathcal{T}} a_i u_{is}^t + \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} (\lambda_{is}^{kt} + \frac{1}{2}\delta - \delta \hat{y}_i^{kt}) y_{is}^{kt} \Bigg) + \theta, \tag{5.41}$$

where

$$\theta = \sum_{s\in\mathcal{S}} \rho_s \Bigg(\frac{1}{2} \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \delta (\hat{y}_i^{kt})^2 - \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \lambda_{is}^{kt} \hat{y}_i^{kt} + \frac{1}{2} \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \delta (\hat{x}_{ij}^{kt})^2 - \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \mu_{ijs}^{kt} \hat{x}_{ij}^{kt} \Bigg). \tag{5.42}$$

Notice that for a given solution for the variables $\hat{y}_i^{kt}$ and $\hat{x}_{ij}^{kt}$, the relaxed formulation decomposes by scenario. Thus, for each scenario $s \in \mathcal{S}$, the subproblem takes the form of a deterministic IRP with lost sales and visiting costs, as follows:

$$\min \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \bar{c}_{ijs}^{kt} x_{ijs}^{kt} + \sum_{i\in\mathcal{N}} \sum_{t\in\mathcal{T}} h_i^t I_{is}^t + \sum_{i\in\mathcal{C}} \sum_{t\in\mathcal{T}} a_i u_{is}^t + \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \bar{b}_{is}^{kt} y_{is}^{kt} \tag{5.43}$$

$$\text{s.t.} \quad (5.2)\text{-}(5.4), (5.11)\text{-}(5.13), (5.28)\text{-}(5.33) \text{ and } (5.36)\text{-}(5.37),$$

where $\bar{c}_{ijs}^{kt} = c_{ij} + \mu_{ijs}^{kt} + \frac{1}{2}\delta - \delta \hat{x}_{ij}^{kt}$ and $\bar{b}_{is}^{kt} = \lambda_{is}^{kt} + \frac{1}{2}\delta - \delta \hat{y}_i^{kt}$ are the routing and visiting costs of the scenario subproblem, respectively.

To devise a PH-based solution method from the previously applied decomposition, we must define a procedure to set the reference solution ($\hat{y}_i^{kt}$ and $\hat{x}_{ij}^{kt}$) as well as a procedure to guide the scenario solutions to a consensus among the first-stage solutions. These procedures are described in the upcoming sections.

5.4.2 Setting the reference solution

In this phase of the method, we use the first-stage solution of each scenario to identify a global trend among them. In our heuristic, we use an aggregation operator that combines the first-stage solutions over all the scenarios into a single solution by taking the weighted sum of each first-stage variable, where the weights are defined by the probability of occurrence of each scenario. This type of aggregation operator was originally proposed by Rockafellar and Wets (1991) and later used by Crainic et al. (2011) and Lamghari and Dimitrakopoulos (2016). Let v define the index of the outer iterations of the PH-based heuristic method. Then, the value of the reference solution variables $\hat{y}_i^{kt(v)}$ and $\hat{x}_{ij}^{kt(v)}$ in the iteration v of the algorithm, are obtained using equations (5.44) and (5.45), respectively, as follows:

$$\hat{y}_i^{kt(v)} = \sum_{s \in \mathcal{S}} \rho_s \bar{y}_{is}^{kt(v)} \qquad \forall i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{5.44}$$

$$\hat{x}_{ij}^{kt(v)} = \sum_{s \in \mathcal{S}} \rho_s \bar{x}_{ijs}^{kt(v)} \qquad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{5.45}$$

where $\bar{y}_{is}^{kt(v)}$ and $\bar{x}_{ijs}^{kt(v)}$ are the values of the first-stage variables of the solution of scenario $s \in \mathcal{S}$ in the v-th iteration of the algorithm.

Notice that when we obtain $\hat{y}_i^{kt(v)}(\hat{x}_{ij}^{kt(v)}) \in \{0,1\}$ for a given node $i \in \mathcal{N}$ (arc $(i,j) \in \mathcal{A}$), vehicle $k \in \mathcal{K}$ and time period $t \in \mathcal{T}$, it means that we have reached a consensus on the values of the variables $y_{is}^{kt(v)}$ $(x_{ijs}^{kt(v)})$ over all the scenarios in iteration v. Then, if a consensus is obtained for all the first-stage variables, the current set of solutions (one for each scenario) composes a feasible solution for the complete stochastic program. However, most of the time this is not the case and we have $0 < \hat{y}_i^{kt(v)} < 1$ or $0 < \hat{x}_{ij}^{kt(v)} < 1$, implying that the current reference solution is infeasible given the integrality requirements of the first-stage variables. Still, the values of the reference solution can be used to indicate the tendency to visit a customer or the usage of an arc by a given vehicle in a defined time period. For a given node (arc), vehicle and period, values of $\hat{y}_i^{kt(v)}$ $(\hat{x}_{ij}^{kt(v)})$ close to one indicate that the node (arc) is being visited (traversed) in the period by that vehicle in most of the scenario solutions. Analogously, a value close to zero indicates a tendency toward not visiting the node (traversing the arc).

5.4.3 Adjustment strategy

In each iteration of the PH-based heuristic, it is necessary to adjust the scenario subproblem costs with the aim of leading to a gradual consensus of the first-stage solutions over all the scenario subproblems and, as a consequence, of the reference solution variables. For this, different strategies can be used such as updating the multipliers λ_{is}^{kt} and μ_{ijs}^{kt} that appear in the routing and visiting costs of the objective function (5.43) of the scenario subproblems, using the augmented Lagrangean method (Bertsekas, 1982; Luenberger and Ye, 2016). Another strategy that can be applied is to use a heuristic rule in each iteration to directly modify the routing and visiting costs ($\bar{c}_{ijs}^{kt}$ and $\bar{b}_{is}^{kt}$, respectively) of the scenarios, instead of the multipliers (Crainic et al., 2011). In our implementation we used a heuristic strategy since it resulted in slightly better results in most of the cases we tested. Two types of heuristic adjustments are applied, namely, a global adjustment to guide the overall search and a local adjustment to influence the search at each scenario subproblem.

Given a reference solution $\hat{y}_i^{kt(v)}$ and $\hat{x}_{ij}^{kt(v)}$ in iteration v, the global adjustment tries to identify trends among the scenario solutions and then sets the costs accordingly. When a low value of $\hat{y}_i^{kt(v)}$ ($\hat{x}_{ij}^{kt(v)}$) is reached, it means that in most of the scenario solutions in iteration v, the node i is not visited (the arc (i, j) is not traversed) by vehicle k in time period t, while a large value of $\hat{y}_i^{kt(v)}$ ($\hat{x}_{ij}^{kt(v)}$) indicates the opposite. Thus, when the value of $\hat{y}_i^{kt(v)}$ ($\hat{x}_{ij}^{kt(v)}$) is less than a given parameter ε^y (ε^x) $\in (0, 0.5)$ we increase the value of the visit (travel) cost with the aim of discouraging the visit to the customer (the use of the arc) in the next iteration by all the scenarios. Similarly, when the value of $\hat{y}_i^{kt(v)}$ ($\hat{x}_{ij}^{kt(v)}$) is greater than $1 - \varepsilon^y$ $(1 - \varepsilon^x)$ we reduce the corresponding parameter so that the visit (usage of the arc) is encouraged in the scenario solutions. This strategy, for the visit and travel costs, is defined in equations (5.46) and (5.47), respectively:

$$\bar{b}_i^{kt(v)} = \begin{cases} \beta \bar{b}_i^{kt(v-1)} & \text{if } \hat{y}_i^{kt(v-1)} < \varepsilon^y, \\ \frac{1}{\beta} \bar{b}_i^{kt(v-1)} & \text{if } \hat{y}_i^{kt(v-1)} > 1 - \varepsilon^y, \\ \bar{b}_i^{kt(v-1)} & \text{otherwise;} \end{cases} \tag{5.46}$$

$$\bar{c}_{ij}^{kt(v)} = \begin{cases} \beta \bar{c}_{ij}^{kt(v-1)} & \text{if } \hat{x}_{ij}^{kt(v-1)} < \varepsilon^x, \\ \frac{1}{\beta} \bar{c}_{ij}^{kt(v-1)} & \text{if } \hat{x}_{ij}^{kt(v-1)} > 1 - \varepsilon^x, \\ \bar{c}_{ij}^{kt(v-1)} & \text{otherwise,} \end{cases} \tag{5.47}$$

with $\beta > 1$, where β is the adjustment rate of the costs.

The local adjustment strategy is applied at the level of each scenario $s \in \mathcal{S}$. We try to identify variables of the scenario solution for which there are large differences w.r.t. the current reference solution and adjust their costs, using equations (5.48) and (5.49) for the visit and routing costs, respectively:

$$\bar{b}_{is}^{kt(v)} = \begin{cases} \beta \bar{b}_i^{kt(v)} & \text{if } |\bar{y}_{is}^{kt(v-1)} - \hat{y}_i^{kt(v-1)}| > M^y \text{ and } \bar{y}_{is}^{kt(v-1)} = 1, \\ \frac{1}{\beta} \bar{b}_i^{kt(v)} & \text{if } |\bar{y}_{is}^{kt(v-1)} - \hat{y}_i^{kt(v-1)}| > M^y \text{ and } \bar{y}_{is}^{kt(v-1)} = 0, \\ \bar{b}_i^{kt(v)} & \text{otherwise;} \end{cases} \tag{5.48}$$

$$\bar{c}_{ijs}^{kt(v)} = \begin{cases} \beta \bar{c}_{ij}^{kt(v)} & \text{if } |\bar{x}_{ijs}^{kt(v-1)} - \hat{x}_{ij}^{kt(v-1)}| > M^x \text{ and } \bar{x}_{ijs}^{kt(v-1)} = 1, \\ \frac{1}{\beta} \bar{c}_{ij}^{kt(v)} & \text{if } |\bar{x}_{ijs}^{kt(v-1)} - \hat{x}_{ij}^{kt(v-1)}| > M^x \text{ and } \bar{x}_{ijs}^{kt(v-1)} = 0, \\ \bar{c}_{ij}^{kt(v)} & \text{otherwise,} \end{cases} \tag{5.49}$$

where the parameters $M^y \in (0, 1)$ and $M^x \in (0, 1)$ represent the threshold defining when a local adjustment has to be applied for the visit and routing variables, respectively.

5.4.4 Solving the IRP(s)

In our solution approach, for each scenario and iteration we have to solve a subproblem corresponding to a deterministic IRP with visiting costs. As pointed out by Crainic et al. (2011), it is not necessary to solve the scenario subproblems to optimality since we are using the PH algorithm as a heuristic procedure. Thus, to solve these problems we use an ILS-based hybrid heuristic, which has been successfully applied to solve other variants of the IRP (Alvarez et al., 2018, 2020). In this method, several components manage the different decisions of the problem. First, the local search heuristic of the method is responsible for the improvement of the routing decisions. This is done using a randomized variable neighborhood descent heuristic. Secondly, a multi-operator procedure, which is used in the perturbation mechanism, handles the visit decisions. This procedure modifies several parts of the input solution every time it is applied. In addition, the method uses a linear programming (LP) model to compute the optimal values of the delivery, inventory and recourse decisions. The input required by this model is a solution given by a set of visit decisions.

5.4.5 Obtaining a feasible solution for the SIRP

Once every scenario subproblem has been solved, we can use their solutions to obtain a feasible solution for the complete problem. For this, notice that we can have three cases for the solutions of the scenario subproblems in iteration v, as follows:

(i) We have a consensus on the first-stage solutions over all the scenarios, i.e., $\hat{y}_i^{kt(v)} \in \{0,1\}$, $\forall i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}$ and $\hat{x}_{ij}^{kt(v)} \in \{0,1\}, \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}$;

(ii) We have a consensus on the visit variables over all the scenarios but we have not reached a consensus on the vehicle flow variables, i.e., $\hat{y}_i^{kt(v)} \in \{0,1\}$, $\forall i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}$ and $0 < \hat{x}_{ij}^{kt(v)} < 1$, for some $(i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}$;

(iii) We have not reached a consensus neither on the visit variables nor on the vehicle flow variables, i.e., $0 < \hat{y}_i^{kt(v)} < 1$, for some $i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}$.

In the first case, we have obtained a feasible solution for the complete multi-scenario problem where the first-stage solution corresponds to the reference solution, i.e., $y_i^{kt} = \hat{y}_i^{kt(v)}$ and $x_{ij}^{kt} = \hat{x}_{ij}^{kt(v)}$, $\forall k \in \mathcal{K}, t \in \mathcal{T}, i \in \mathcal{N}$ and $(i,j) \in \mathcal{A}$, respectively. In the second case, the scenario solutions visit the same customers using the same vehicles in the same time periods but in different orders (using different vehicle routes). In this case, we can take the scenario solution with the lowest first-stage cost and use the values of its first-stage solution ($\bar{y}$ and $\bar{x}$) to obtain a feasible solution for the complete problem by solving the following LP model:

$$\min \quad \sum_{s \in \mathcal{S}} \rho_s \left(\sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} h_i^t I_{is}^t + \sum_{i \in \mathcal{C}} \sum_{t \in \mathcal{T}} a_i u_{is}^t \right) \tag{5.50}$$

$$\text{s.t.} \quad q_{is}^{kt} \leq \min\{Q, C_i\} \bar{y}_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.51}$$

$$\sum_{i \in \mathcal{C}} q_{is}^{kt} \leq Q \bar{y}_0^{kt} \qquad k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}, \tag{5.52}$$

(5.2)-(5.4) and (5.11)-(5.13),

where $\bar{y}_0^{kt} = 1$ indicates that vehicle k is used in time period t and $\bar{y}_i^{kt} = 1$ indicates that vehicle k visits customer i in time period t. The objective function (5.50) consists of minimizing the total cost, given by the sum of inventory holding and penalty costs. Constraints (5.51) link delivery and visit variables. Finally, constraints (5.52) impose the capacity of each vehicle.

In the last case, when there is no consensus on the visit variables, and consequently on the vehicle flow variables, we can still use a scenario first-stage solution to obtain a feasible solution for the whole problem. In our implementation we use the solution of scenario $\bar{s}$, such that, $\bar{s} = \arg\max_{s\in\mathcal{S}}\{\sum_{i\in\mathcal{C}} \bar{a}_i \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} \bar{y}_{is}^{kt}\}$, where $\bar{a}_i = a_i/\max_{j\in\mathcal{C}}\{a_j\}$, i.e., we take the solution with the largest value of the weighted number of visits. Using the first-stage solution of $\bar{s}$ ($\bar{y}$ and $\bar{x}$) we can obtain a feasible solution for the complete problem by solving the same LP model described for the second case. It is worth mentioning that for this case we tried different strategies for the selection of the first-stage scenario solution used as basis to generate a solution for the multi-scenario problem and the above mentioned criterion led to slightly better results.

In addition, we try to improve the solution found by applying an ILS-based hybrid method similar to the one described in Section 5.4.4 but in this case for the multi-scenario problem. Thus, the local search and perturbation components are the same as before, and the LP model is replaced by a multi-scenario LP model.

5.4.6 Description of the complete heuristic

Given the components described in the previous sections, we can now describe the general structure of the solution method, whose pseudo-code is shown in Algorithm 5.1. Each outer iteration corresponds to an iteration of the main loop (lines 3 to 16). In the first outer iteration of the method (line 4), we set the initial values of the scenario subproblem costs (lines 5 to 6). For subsequent iterations, the required global and local adjustment strategies are applied (lines 8 and 9, respectively) at the beginning of the outer iterations. After that, we solve every scenario subproblem (line 11) and compute the updated reference solution of the iteration (line 12 and 13). At the end of every outer iteration, we construct a feasible solution for the problem (line 14) and update the best feasible solution if a new one was found. In the second outer iteration of the heuristic ($v = 1$), the visiting costs are set to an initial value b_0 in the local adjustment strategy (line 9), i.e., $b_{is}^{kt(v)} = b_0,\ \forall i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}$ when $v = 1$. This is done because in the original version of the problem (before the scenario decomposition) there are no visiting costs. The algorithm stops when either a consensus is reached over all the first-stage variables, when it reaches the maximum number of outer iterations, or when the running time limit is exceeded (line 3).

5.5 Computational experiments

In this section, we report the results obtained with the formulations and the heuristic algorithm previously presented. This algorithm was coded in C++ and run on a 2.1 GHz AMD Opteron

Algorithm 5.1: PH-based heuristic

begin
  $v = 0$;
  while *stopping criterion is not met* **do**
    if $v = 0$ **then**
      $\bar{b}_{is}^{kt0} = 0 \quad \forall i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}$;
      $\bar{c}_{ijs}^{kt0} = c_{ij} \quad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}$;
    else
      Apply global adjustment strategy;
      Apply local adjustment strategy $\forall s \in \mathcal{S}$;
    end
    for $s \in \mathcal{S}$ **do** Solve IRP(s) ;
    $\hat{y}_i^{kt(v)} = \sum_{s \in \mathcal{S}} \rho_s \bar{y}_{is}^{kt(v)} \qquad \forall i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}$;
    $\hat{x}_{ij}^{kt(v)} = \sum_{s \in \mathcal{S}} \rho_s \bar{x}_{ijs}^{kt(v)} \qquad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}$;
    Apply feasible solution generation procedure;
    $v = v + 1$;
  end
end

6172 processor with one thread and a limit of 18 GB of RAM. We used CPLEX v12.8 as solver. We turned off CPLEX's parallel mode and set the CPLEX optimality tolerance to 10^{-6}. All other CPLEX parameters were set to their default values.

5.5.1 Test instances

To test our algorithms, we generated problem instances based on the benchmark set proposed by Archetti et al. (2007) for the deterministic basic variant of the IRP with a single vehicle. The original set is divided into four sets: H3, L3, H6, and L6, where L (H) stands for low (high) inventory holding costs while the digit (3 or 6) indicates the number of time periods for the instances of the set. For our experiments, we used sets H3 and H6 only as these instances provide a more significant trade-off between routing and inventory holding costs. Also, we created a new set (H9) from the instances of set H6 by extending the planning horizon of these instances to nine time periods (instead of six). It is worth mentioning that in the instances of Archetti et al. (2007), the value of the customer demands ($\bar{d}_i^t$) and the amounts made available at the supplier ($\bar{r}^t$) are constant over the planning horizon, i.e., $\bar{d}_i^t = \bar{d}_i$, $\forall i \in \mathcal{C}$, $t \in \mathcal{T}$ and $\bar{r}^t = \bar{r}$, $\forall t \in \mathcal{T}$, respectively.

We changed some parameter values of the instances with the aim of increasing the sensitivity to the random variables. Specifically, we reduced the amounts received by the supplier in each period by 25%, set the initial inventory at each customer to $\bar{I}_i^0 = 0.1T\bar{d}_i$, $\forall i \in \mathcal{C}$, and set the initial inventory at the supplier to $\bar{I}_0^0 = 0.1T \sum_{i \in \mathcal{C}} \bar{d}_i$. These changes were performed given that, for some customers, the original value of the initial inventory was large enough to serve the demand for up to two time periods, whereas the supplier initial inventory was large enough to cover all the customer demands, concealing the impact of the uncertain parameters. On the other hand, to increase the relative importance of the inventory management on the total cost we multiplied the inventory holding costs by 5.

The supply and demand were assumed to be independent random variables and the scenarios were generated using a discrete uniform distribution applying a Monte Carlo simulation. For the supply, we used the range $[\bar{r}^t(1-\epsilon^r), \bar{r}^t(1+\epsilon^r)]$, $\forall t \in \mathcal{T}$, where $\epsilon^r \in [0,1]$ is the supply uncertainty level. For the demand, we used the range $[\bar{d}_i^t(1-\epsilon^d), \bar{d}_i^t(1+\epsilon^d)]$, $\forall i \in \mathcal{C}, t \in \mathcal{T}$, where $\epsilon^d \in [0,1]$ is the demand uncertainty level. We set the probability of occurrence of each scenario as $\rho_s = 1/S$, $\forall s \in \mathcal{S}$, i.e., all the scenarios have the same probability of occurrence.

It is worth highlighting that, with the changes applied to the instance parameters and the assumed distributions for the uncertain parameters, the expected demand coverage is 95%, i.e., the expected value of the sum of the amounts made available at the supplier plus the initial inventories ($\mathbb{E}[\sum_{t\in\mathcal{T}} r^t] + I_0^0 + \sum_{i\in\mathcal{C}} I_i^0$) equals $0.95D$, where D is the expected value of the sum of the demands of all the customers over the planning horizon ($D = \mathbb{E}[\sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}} d_i^t]$). This applies for all the instances, independent of the length of the planning horizon.

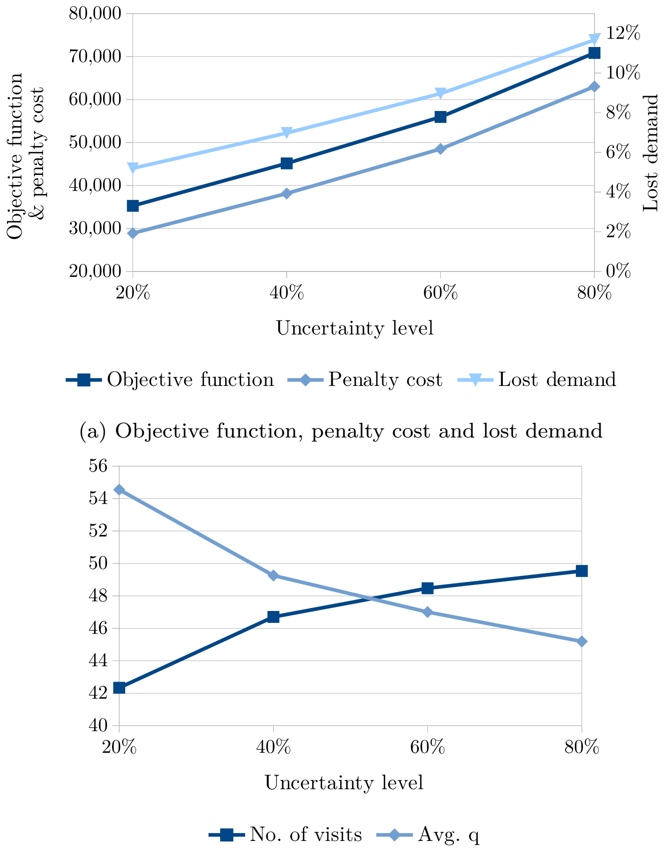
The values of the penalty terms were set as $a_i = \lceil \hat{a}(F + 2c_{0i}/m^c) \rceil$, with $m^c = \max_{i\in\mathcal{C}}\{c_{0i}\}$ and where $\hat{a}$ is a predefined penalty level and F is a fixed penalty value, as in Adulyasak et al. (2015a). Unless stated otherwise, we use $\hat{a} = 4$ and $F = 50$ for the lost sales and backlogging penalties and $\Delta = 0.1\bar{r}$, $f = 5{,}000$ and $p = \hat{a}F/2$ for the CRC parameters.

In our experiments, we considered the instances with up to 30 customers. Travel costs are computed as Euclidean distances and then rounded to the nearest integer. We use one and two vehicles, dividing the vehicle capacity of the original instance by the number of vehicles and then rounding to the nearest integer.

5.5.2 Results with the lost sales formulation

This section shows the results of the formulation with lost sales as recourse, presented in Section 5.3.1. We imposed a maximum running time of two hours to solve each instance with the formulation. We solved all the instances with 100, 200 and 500 scenarios and for increasing values of the uncertainty levels ϵ^r and ϵ^d, from 0.2 to 0.8 (increasing both supply and demand uncertainty by 0.2), but only show the results for those instances solved to optimality within the time limit. Figure 5.3a shows the average objective function value, penalty cost and percentage of lost demand of all the optimal solutions, and Figure 5.3b indicates the average total number of visits (No. of visits) and delivery size (Avg. q). In addition, Table 5.1 shows the average time (Time) in seconds required to solve all the instances with the given uncertainty levels ($\epsilon^r = \epsilon^d$), the value of the stochastic solution (VSS/OF) and the expected value of perfect information (EVPI/OF) (Birge and Louveaux, 2011) w.r.t. the objective function value. In the table, columns 1 to 3 display the number of time periods (T), vehicles (K) and scenarios (S) of the instances, respectively. Column 4 (#) shows the number of instances solved to optimality (out of 30) for the instances with the sizes given in columns 1 to 3. Each cell shows the average value of the respective column header over all the instances solved to optimality (#). The results include only the instances solved to optimality in all the four cases of the uncertainty levels (i.e., $\epsilon^r = \epsilon^d = 0.2, 0.4, 0.6, 0.8$), in order to be able to compare the results.

VSS is used to compare the stochastic solution with an expected value approach. It measures the potential gains when solving the stochastic problem instead of a simple expected value problem (EV), in which the random variables are replaced by their expected values in a single scenario approach. To compute the VSS, one has to solve a problem calculating the expected result of using the EV solution (EEV), which is the same stochastic programming model in which its first-stage variables are fixed to the values of the solution of the EV problem. VSS is computed as the difference between the optimal values of the EEV problem and the stochastic programming model (RP), i.e., VSS = EEV − RP. EVPI is used to compare the stochastic solution with the wait-and-see approach, in order to provide a measure of how good a solution would be under perfect information. EVPI is computed as the difference between the optimal values of the stochastic programming model and the expected wait-and-see solution (WS), i.e., EVPI = RP − WS, where WS = $\sum_{s \in \mathcal{S}} \rho_s W^*(s)$ and $W^*(s)$ is the optimal value of the single-scenario deterministic problem associated with scenario $s \in \mathcal{S}$. We set a time limit of two hours to compute the VSS and EVPI values.



(a) Objective function, penalty cost and lost demand

(b) Number of visits and average delivery size

Figure 5.3: Behavior of the solutions for the lost sales case for increasing uncertainty levels

As expected, these results show that increasing uncertainty levels results in larger values of the objective function, as a consequence of the larger values of the lost demand. Also, when we increase the uncertainty levels, the solutions tend to perform more visits and the delivery sizes tend to decrease. This can be viewed as a protection mechanism against the

Instance set				Time (secs)				VSS/OF (%)				EVPI/OF (%)			
T	K	S	#	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
3	1	100	30	79.46	14.28	7.77	5.94	21.22	30.28	29.00	29.19	1.89	2.02	2.11	1.80
6	1	100	20	1,388.16	100.66	37.12	16.65	16.31	27.92	36.88	36.99	2.39	2.87	2.81	2.56
9	1	100	7	919.72	115.09	48.39	17.43	13.03	28.93	33.89	41.76	2.70	4.63	4.05	3.96
3	2	100	19	1,145.68	210.43	183.52	160.94	24.10	31.35	30.39	29.08	2.78	3.70	3.74	3.64
6	2	100	5	1,950.46	550.97	465.71	321.28	14.89	30.00	33.08	34.35	2.90	4.71	4.95	5.13
3	1	200	30	258.48	31.20	19.21	12.51	21.52	29.90	29.09	29.19	1.92	2.16	2.12	1.86
6	1	200	16	2,093.92	222.19	81.76	39.53	16.47	27.41	35.30	37.54	2.49	2.95	3.20	3.09
9	1	200	6	3,160.64	468.82	132.81	52.77	10.65	28.23	32.50	38.69	2.61	4.12	4.21	4.10
3	2	200	15	1,368.06	594.36	322.03	185.53	21.77	26.21	30.20	29.06	3.18	4.09	4.18	3.90
6	2	200	2	5,566.76	1,397.09	593.60	205.71	17.44	26.53	35.95	38.46	3.71	4.99	5.53	5.67
3	1	500	30	1,383.54	196.65	87.28	51.42	21.41	29.78	30.38	29.06	1.70	1.91	1.85	1.64
6	1	500	6	1,531.68	362.76	127.61	68.47	12.02	22.10	27.54	28.13	2.78	3.89	3.61	3.62
3	2	500	8	1,753.16	1,273.75	1,026.23	756.35	25.44	27.03	27.03	28.83	2.96	4.50	4.35	4.22
Avg.			194	1,158.26	243.71	141.35	91.86	19.79	28.91	31.12	31.57	2.29	2.87	2.88	2.68

Table 5.1: Results for the lost sales formulation for increasing values for ϵ^r and ϵ^d

uncertainties of the problem, which also results in larger values of the total transportation cost. Increasing uncertainty levels also resulted in larger relative VSS. For uncertainty levels of 20% ($\epsilon^r = \epsilon^d = 0.2$), the average VSS represents approximately 20% of the objective function value while for 80% of uncertainty levels ($\epsilon^r = \epsilon^d = 0.8$) it represents more than 30% of this value. These results justify the use of the stochastic programming model. Regarding the EVPI, notice that they are relatively small when compared to the objective function value. This implies that, even if the supplier had a perfect forecast for the supply and demand, this would not result in significantly better solutions, which shows the robustness of the solutions obtained with the stochastic model.

From these results, it is also possible to observe the difficulty of solving the formulation using a general-purpose optimization software, given that only 194 of the considered instances (540 in total) were solved to optimality for the four cases of the uncertainty levels. This difficulty increases especially with growing values of the number of time periods (T) and vehicles (K). Notice also that the time the solver takes to solve the instances decreases with increasing values of the uncertainty levels. This can be partially explained by the fact that dominant scenarios (i.e., scenarios with low supply and large customer demands) are more likely to appear with larger uncertainty levels. These scenarios condition the first-stage decisions since the lost sales costs are generally high. Thus, the optimal solutions will tend to avoid out-of-stock situations by planning more visits even though the product might not be available at the supplier. This is related to the protection mechanism previously mentioned. It is worth remembering that the number displayed in column '#' corresponds to the number of instances solved to optimality in all the four cases of the uncertainty levels. Notice also that the table displays fewer cases (13 rows) than the total explored ($18 = 3 \times 2 \times 3$) for the combinations of the values for T, K and S, since for some combinations the solver could not solve to optimality any of the instances for the four explored uncertainty level values.

In a different analysis, for the solutions of the instances that were solved optimally within the time limit when considering uncertainty in both supply and demand ($\epsilon^r = \epsilon^d = 0.6$), only in supply ($\epsilon^r = 0.6, \epsilon^d = 0.0$), and only in demand ($\epsilon^r = 0.0, \epsilon^d = 0.6$), Figure 5.4 shows the average objective function value and total penalty cost, Figure 5.5 displays the average total number of routes (No. of routes) and visits (No. of visits), and Figure 5.6 indicates the average VSS and EVPI w.r.t. the objective function value. It can be observed that smaller values of the objective function and penalty cost are obtained when we consider either only uncertain supply or only uncertain demand, when compared to the case when both are uncertain (Figure 5.4). This is a result of the reduced number of uncertain parameters in the former cases. The results also show that supply uncertainty has a larger impact on the lost sales values, compared to the case with only demand uncertainty. This may be due to the pooling effect of the demand uncertainty for each customer within each time period, i.e., the variability of the demand in each time period is partially canceled among the demands themselves. This results in a larger number of routes and visits for the case when only the product supply is uncertain compared to the case when only the demands are uncertain (Figure 5.5). Again, large relative VSS justify the use of the stochastic programming model even for the cases in which only one of the parameters is uncertain. Notice that, analogously to the previous experiment (with increasing uncertainty levels), the relative EVPI is relatively small, showing the robustness of the solutions of the stochastic model (Figure 5.6).

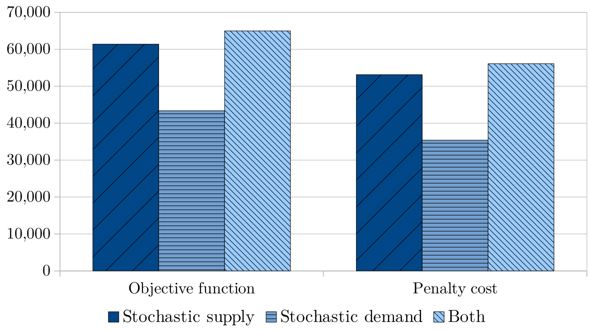


Figure 5.4: Objective function and penalty cost

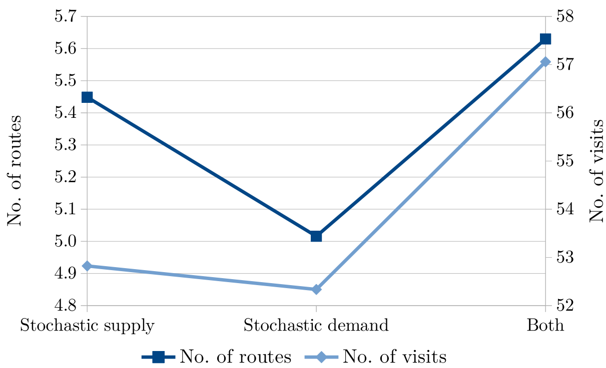


Figure 5.5: Number of routes and visits

Figure 5.7 displays the average value of the objective function for the cases with just uncertain supply or just uncertain demand, as a percentage of the value when both parameters are uncertain. We used the results of the instances with three time periods, one vehicle and 200

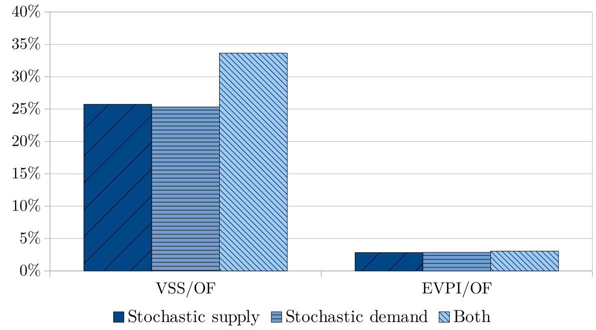


Figure 5.6: Relative VSS and EVPI

scenarios and show the average value over the five instances with the indicated size, separated by the different number of customers, from 5 to 25 customers (we omitted the results with 30 customers since one of the instances was not solved to optimality within the time limit). The aim of this experiment was to analyze the pooling effect of the customer demands in relation to the number of customers considered. In the figure, it can be observed that the values for the case with just demand uncertainty tend to decrease as we increase the number of customers, getting closer to the case in which all the parameters are deterministic. This may be explained by the pooling effect as it is more likely that the uncertain demands compensate each other when we consider a larger number of customers. On the other hand, the values for the case with just uncertain supply tend to increase when the number of customers grows. This may be explained by the fact that when more customers are considered then more lost demand is to be expected given the uncertain nature of the supply.

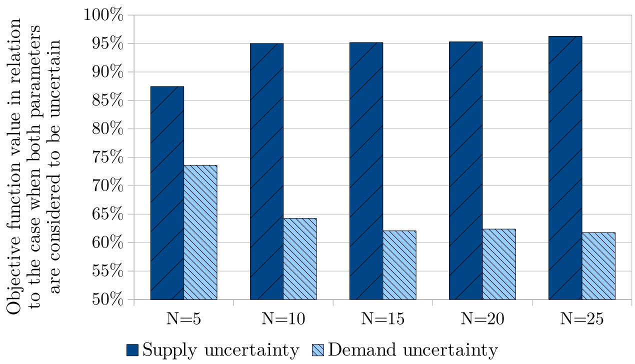


Figure 5.7: Results with either stochastic supply or stochastic demands for different numbers of customers

5.5.3 Results with the backlogging formulation

This section shows the results with the formulation using backlogging as the recourse in case of out-of-stock situations, as presented in Section 5.3.2. The objective is to show the differences between the solutions with this recourse and the ones with lost sales as the recourse. For this, we set an uncertainty level of 60% for both supply and demand (i.e., $\epsilon^r = \epsilon^d = 0.6$) and imposed a time limit of two hours to solve each instance with both formulations. We used the same supply and demand realizations for both formulations and we used 100, 200 and 500 scenarios. We also used the same values for the backlogging and lost sales costs (a_i). For comparison purposes, we only analyze the results for those instances solved to optimality within the time limit with both formulations for the given uncertainty level (329 instances). For all the solutions in both cases, Figure 5.8 shows the average objective function and penalty cost (recourse) values, while Figure 5.9 displays the VSS and EVPI w.r.t. the objective function value. As expected, since in the backlogging case the unserved demands accumulate from one period to the other, the average backlogged amounts are considerably higher than the average lost sales (in the lost sales case). This results in larger penalty values and, consequently, in larger values of the objective function (Figure 5.8). Also, notice that the relative VSS and EVPI are smaller for the backlogging case than for the lost sales case (Figure 5.9), which may be explained by the fact that in the backlogging case there are larger values of the objective function. However, for the backlogging case, the average VSS still represents more than 20% of the objective function value, which also justifies the use of the stochastic programming model. Similarly to the lost sales case, the EVPI is relatively small when compared to the objective function value (Figure 5.9). This may imply that even if the supplier had the perfect supply and demand forecast, this would not result in significantly better solutions, showing the robustness of the solutions obtained with the stochastic model.

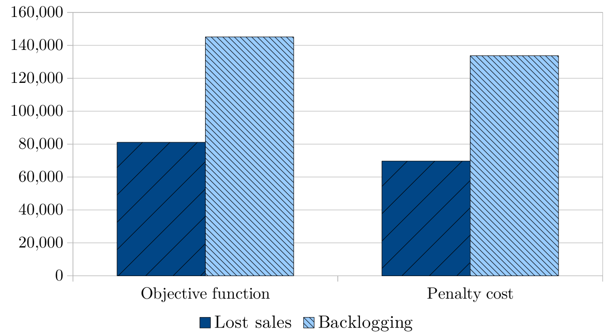


Figure 5.8: Objective function and penalty cost

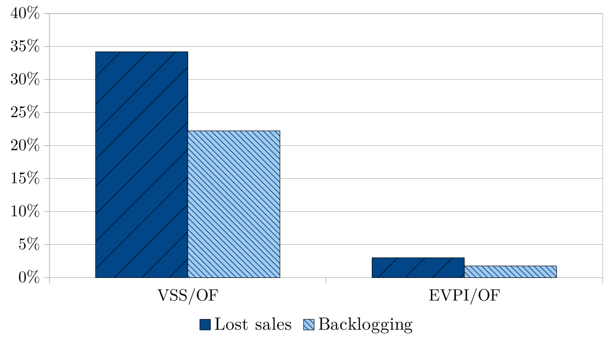


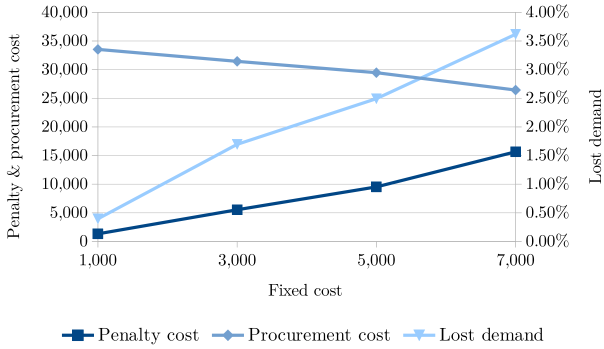
Figure 5.9: Relative VSS and EVPI

Similarly to the lost sales case, when backlogging is used as recourse in the second stage, we could also observe the protection mechanism consisting of performing more visits with smaller delivery quantities for increasing values of the uncertainty levels (for both ϵ^r and ϵ^d). For the sake of brevity we do not display these results. It is also worth mentioning that the average CPU time to solve the considered instances was 467 seconds with the backlogging formulation, compared to 811 seconds with the lost sales case recourse. It is also worth mentioning that the average VSS obtained for the lost sales case in this experiment is larger than that observed in the previous experiment and shown in Table 5.1. This is a result of the subset of instances that are used for comparison purposes, since in each experiment we consider the results for those instances solved to optimality within the time limit for the analyzed cases.

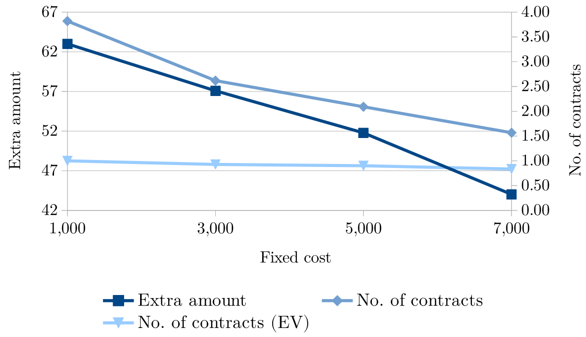
5.5.4 Results with the CRC and lost sales formulation

This section shows the results for the CRC formulation with lost sales, presented in Section 5.3.3. We analyze the results for increasing values of the fixed contracting cost, namely $f \in \{1{,}000; 3{,}000; 5{,}000; 7{,}000\}$. For this, we set an uncertainty level of 60% for both supply and demand (i.e., $\epsilon^r = \epsilon^d = 0.6$) and imposed a time limit of two hours to solve each instance. We used the same supply and demand realizations for each value of the fixed cost and, as in the previous experiments, we used 100, 200 and 500 scenarios. For comparison purposes, we only analyze the results for those instances solved to optimality within the time limit for all the four values used for the fixed cost (317 instances). Figure 5.10a shows the average values of the penalty (lost sales penalty) and procurement cost as well as the average lost demand of the solutions; while Figure 5.10b displays the average extra amount acquired in each period and number of contracts; and Figure 5.10c indicates the VSS and EVPI w.r.t. the objective function value. Additionally, we show in Figure 5.10b the average number of contracts in the EV solutions, computed as part of the VSS.

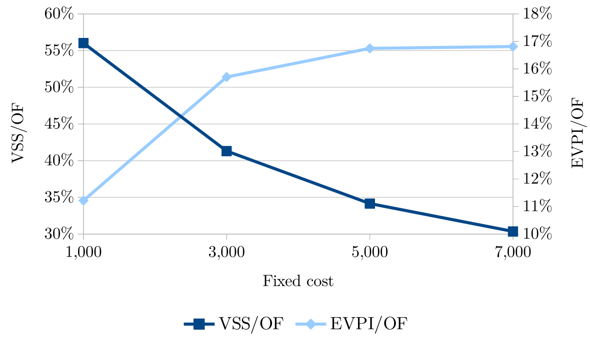
The results show that as we increase the values of the fixed contracting cost, fewer contracts are reserved by the supplier (thus, less extra amount is acquired) (Figure 5.10b), which results in considerably larger values of the lost demand (and then in larger penalties) (Figure 5.10a). This observation highlights the trade-off between the cost of the lost demand, and the procurement and delivery cost of the extra amounts. Figure 5.10c shows that increasing values of the fixed contracting cost result also in larger relative EVPI, as a consequence of the increasing total cost of the solution of the stochastic model. On the other hand, the relative VSS decreases when we increase f. This latter observation implies that a large value for the fixed contracting cost does not compensate the penalties incurred by lost sales, thus making the model less reactive against uncertainties, which reduces the gains of the stochastic programming formulation. Also, in Figure 5.10b shows that the solutions using an expected value for the stochastic parameters (EV solution) tend to underestimate the number of contracts required, which leads to large lost demand values. The last observation highlights the advantages of the stochastic programming model and the potential consequences of ignoring the stochastic nature of the parameters in the problem.



(a) Penalty cost, procurement cost and lost demand



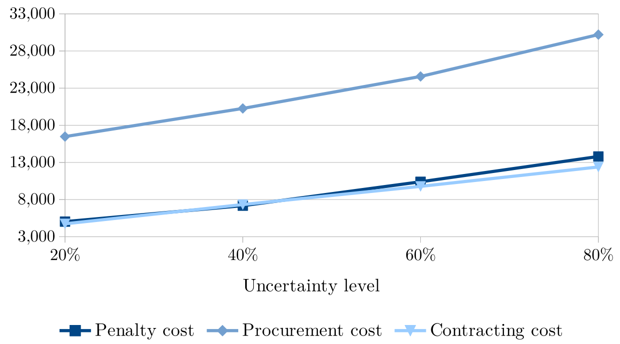
(b) Extra amount acquired and number of contracts



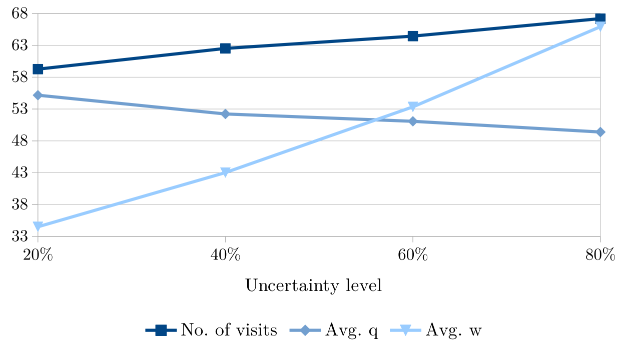
(c) Relative VSS and EVPI

Figure 5.10: Results with the CRC and lost sales formulation for different contracting costs

We also analyzed the results with this formulation for increasing values of the uncertainty levels ϵ^r and ϵ^d, from 0.2 to 0.8 (increasing both supply and demand uncertainty by 0.2), for a value of the fixed contracting cost (f) of 5,000. We imposed a maximum running time of two hours to solve each instance with the formulation. We solved all the formulations with 100, 200 and 500 scenarios, but only show the results for those instances solved to optimality within the time limit for the four cases of the uncertainty levels. Figure 5.11a displays the average values of the penalty (lost sales penalty), procurement and contracting cost; while Figure 5.11b shows the average total number of visits (No. of visits), delivery size (Avg. q) and extra amount acquired in each period (Avg. w); Figure 5.12a shows the average number of contracts in the stochastic and EV solutions (computed as part of the VSS); and Figure 5.12b indicates the average VSS and EVPI w.r.t. the objective function value. In the charts we can observe that increasing uncertainty levels resulted in larger values of the penalty costs as a consequence of the larger values of the lost demand. This also results in increasing values for the procurement and contracting costs (Figure 5.11a). In this case it is also possible to observe the protection mechanism consisting of planning more visits (and now more contracts) in the first stage and performing smaller deliveries in the second stage (Figure 5.11b). Increasing uncertainty levels resulted also in larger relative VSS and EVPI (Figure 5.12b). It also possible to observe in Figure 5.12a that the number of contracts in the EV solutions remains relatively stable for increasing uncertainty levels while the stochastic solutions adapt to this increase. This observation may partially explain the increasing VSS (Figure 5.12b).



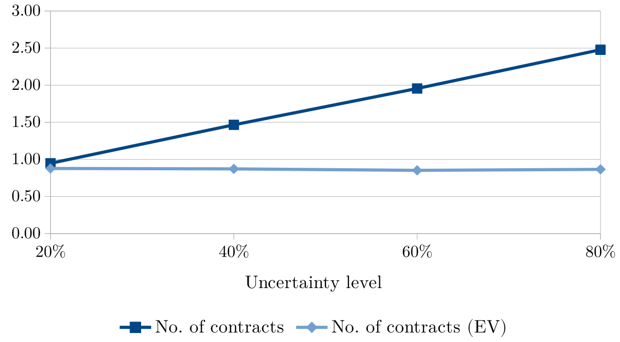
(a) Penalty, procurement and contracting costs



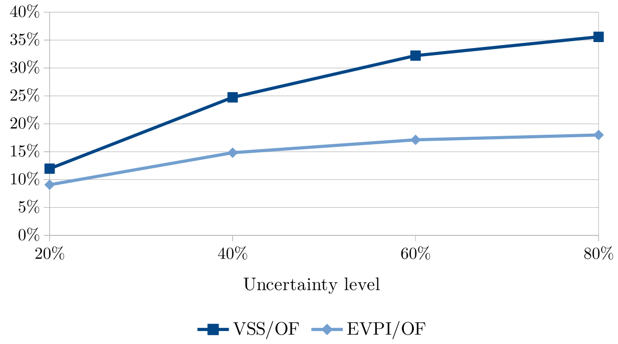
(b) Number of visits, delivery size and extra amount acquired

Figure 5.11: Results with the CRC and lost sales formulation for different uncertainty levels (a)

Figure 5.13 shows the average penalty, procurement and contracting cost for all the instances that were optimally solved within the time limit when considering uncertainty in both supply and demand, only in supply, or only in demand. Analogously to the experiments with the lost sales formulation (Section 5.5.2), it can be observed that smaller values of the penalty cost are obtained when we consider only uncertain demand compared to the case with only uncertain supply. This may again be due to the pooling effect of the customer demands within each time period. As a result, less procurement and contracting costs are incurred in the case with only demand uncertainty when compared to the other two cases.



(a) Number of contracts in the stochastic and EV solutions



(b) Relative VSS and EVPI

Figure 5.12: Results with the CRC and lost sales formulation for different uncertainty levels (b)

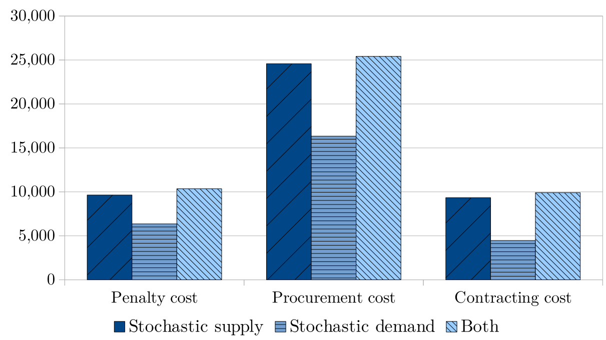


Figure 5.13: Results with the CRC and lost sales formulation with either stochastic supply or demand or both

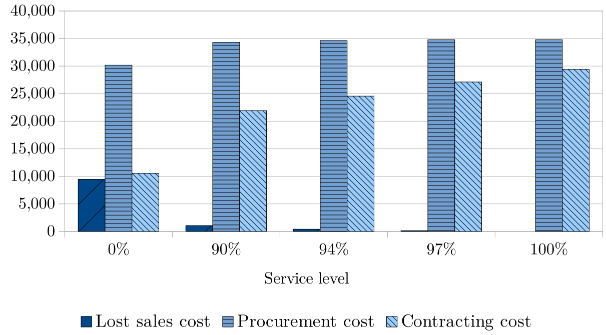
5.5.4.1 Setting a service level in the CRC formulation

In this section we show the results with the CRC and lost sales formulation when we impose a minimum service level. For this, we included constraints of the form

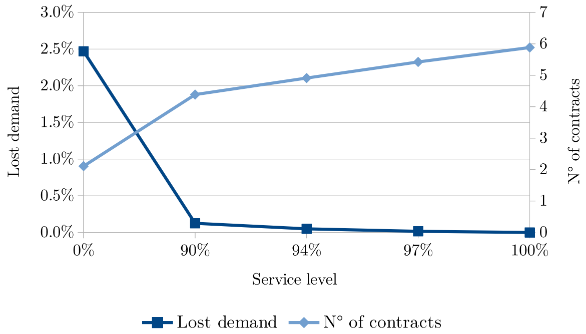
$$\sum_{i \in \mathcal{C}} \sum_{t \in \mathcal{T}} u_i^t \leq (1-\kappa) \sum_{i \in \mathcal{C}} \sum_{t \in \mathcal{T}} d_{is}^t \qquad \forall s \in \mathcal{S} \tag{5.53}$$

to the formulation. These constraints limit the total lost demand in each scenario to a fraction of the scenario total demand, set by the required service level κ. This parameter defines the minimum percentage of the total demand that must be served in each scenario. We considered five cases, four with different service levels (90%, 94%, 97% and 100%) and the base case ($\kappa = 0$). Again, we set an uncertainty level of 60% for both supply and demand (i.e., $\epsilon^r = \epsilon^d = 0.6$) and imposed a time limit of two hours to solve each instance. We compare and analyze the results for those instances solved to optimality within the time limit for all the five values of κ (314 instances). Figure 5.14a shows the average lost sales, procurement and contracting costs of all the considered instances for the respective service level (given in the horizontal axis); and Figure 5.14b shows the average percentage of the demand that is lost and the number of contracts in each case.

As expected, increasing values of the minimum service level (κ) resulted in reduced values for the lost sales cost (and lost demands). This reduction in lost demand is achieved by increasing the number of contracts and the extra amount acquired from the external source, which increases significantly the contracting and procurement costs, respectively. It is worth pointing out that from a value of 90% and higher for the service level, the values of the lost demand decreased considerably (0.1% of lost demand, on average) compared to the case without service level. It is also worth mentioning that there are several different ways of modeling service level requirements for production and distribution systems. For more details we refer to Tempelmeier (2013) and Gruson et al. (2018).



(a) Lost sales, procurement and contracting costs



(b) Lost demand and number of contracts

Figure 5.14: Results with the CRC and lost sales formulation with service level constraints

5.5.5 Results with the progressive hedging-based heuristic

In this section we analyze the performance of the PH-based heuristic approach. We compare its performance with the results obtained with CPLEX with a time limit of two hours. We used three stopping criteria for the heuristic: *i*) a time limit of 600 seconds; *ii*) a maximum of S (number of scenarios) iterations; and *iii*) a maximum of 20 iterations without improvement. We set the values of the parameters of the heuristic to $\varepsilon^y = \varepsilon^x = 0.2$, $M^y = M^x = 0.8$ and $b_0 = 1.0$. For the CRC case we set $\varepsilon^z = 0.5$, $M^z = 1.0$. All these values were determined through preliminary experiments. In addition, the ILS-based hybrid method, used to solve the scenario subproblems and in the feasible solution generation procedure, was configured as in Alvarez et al. (2020, 2018), using as stopping criteria the running time limit of 0.5 and five seconds in the former and latter case, respectively, and the maximum number of iterations, which was set to 1,000.

For the tables in this section, columns labeled with 'CPLEX' show the results of the solver, where '#F' and '#O' show the number of feasible and optimal solutions found by the solver within the time limit, 'Opt gap' shows the optimality gap of the solutions found by the solver and 'Time' shows the running time of the solver (in seconds). Columns labeled with 'PH' show the results with the PH-based heuristic. 'Dif UB' shows the relative difference between the value of the best solutions found by the heuristic (z^h) and the value of the best feasible solutions found by the solver (z^f), computed using the formula $100 \times (z^h - z^f)/z^f$. Column 'Dif LB' shows the optimality gap of the obtained solutions, computed as $100 \times (z^h - \underline{z})/\underline{z}$, where $\underline{z}$ is the best LB computed by the solver. Finally, 'Time' shows the CPU required by the heuristic (in seconds). The values of 'Dif UB' and 'Dif LB' are computed only over those instances for which the solver could find a feasible solution within the two-hour time limit.

First, in Table 5.2, we report the results for the lost sales case, setting a value of 60% for the uncertainty levels for both supply and demand ($\epsilon^r = \epsilon^d = 0.6$). Each line shows the average for all the instances with the dimensions shown in columns 1 to 3. The last row (Avg.) displays shows the mean values of the corresponding column. The results show that on average the heuristic is able to find reasonably good feasible solutions within relatively short CPU times when compared to CPLEX. Specifically, the heuristic can find solutions with an average relative difference of less than 1% compared to those provided by the solver. These solutions have an average optimality gap of 2.34% and were obtained in an average CPU time of 619 seconds, which represents 27% of the time spent by the solver. It is worth highlighting that the heuristic was able to find feasible solutions for all the 540 instances considered in this experiment while the solver could find feasible solutions for only 80% of the instances (430 instances) within the time limit and with an average CPU time of more than 2,200 seconds.

Regarding the specific algorithmic components leading to the final solutions found by the PH-based heuristic, it is worth mentioning that 119 of the final solutions were found in the third case of the feasible solution generation procedure of the heuristic (Section 5.4.5), while the remaining (421 solutions) were found by the ILS-based hybrid heuristic component also included in the same procedure. This result highlights the importance of the feasible solution generation procedure included in our heuristic. The results also reveal the advantages of the heuristic to

Instance set			CPLEX				PH		
					Opt	Time	Dif	Dif	Time
T	K	S	#F	#O	gap (%)	(secs)	UB (%)	LB (%)	(secs)
3	1	100	30	30	0.00	7.77	0.31	0.31	528.92
6	1	100	30	30	0.00	172.83	0.92	0.92	556.27
9	1	100	30	29	0.00	647.21	1.57	1.57	570.75
3	2	100	30	26	0.02	1,822.79	2.01	2.03	550.08
6	2	100	28	10	10.19	5,376.14	-5.75	4.93	565.80
9	2	100	17	5	0.93	6,177.60	5.31	6.30	577.50
3	1	200	30	30	0.00	19.21	0.28	0.28	564.98
6	1	200	30	30	0.00	443.71	1.00	1.00	596.95
9	1	200	30	28	0.00	1,712.54	1.93	1.93	605.43
3	2	200	30	23	0.12	2,853.08	1.74	1.86	597.91
6	2	200	20	6	1.27	5,680.21	4.08	5.43	604.72
9	2	200	11	1	11.50	7,069.93	-4.44	8.60	627.66
3	1	500	30	30	0.00	87.28	0.28	0.28	685.88
6	1	500	29	27	0.01	1,841.89	2.02	2.03	724.73
9	1	500	16	13	0.15	3,286.60	2.69	2.84	698.05
3	2	500	25	14	2.58	4,219.84	-0.16	2.45	740.32
6	2	500	9	2	6.74	6,616.20	-1.23	6.19	698.36
9	2	500	5	0	17.47	7,200.23	-12.29	6.13	647.76
Avg.			430	334	1.56	2,295.26	0.67	2.34	619.00

Table 5.2: Results with the PH-based heuristic for the lost sales case with $\epsilon^r = \epsilon^d = 0.6$

solve larger instances when compared to the solver. For instance, for sets with nine time periods, two vehicles and 200 and 500 scenarios, the heuristic finds solutions that are on average up to 12% better than those found by the solver, which could only find 16 feasible solutions (out of 60 instances) and prove the optimality of one of them. Notice also that for some sets the total time of the heuristic might be larger than 600 seconds. This is due to the fact that we do not stop the execution of the heuristic during inner iterations (i.e., solving the scenario subproblems) but only after outer iterations. It is worth mentioning that the values of '#F' and '#O' in this table do not match those shown in Table 5.1 since in the latter table we display the results only for those instances solved to optimality for all the four cases of the uncertainty levels (Section 5.5.2).

We also analyzed the performance of the heuristic under different uncertainty level values. The results are displayed in Table 5.3. The results show that the heuristic is able to find reasonably good feasible solutions when compared to the solver for different values of the uncertainty levels. In particular, when we consider relatively low uncertainty levels (e.g., $\epsilon^r = \epsilon^d = 0.2$) then the heuristic finds solutions that are on average better than those found by the solver within the time limit. It is also possible to observe the relatively stable behavior of the heuristic in terms of running times for the different uncertainty levels that were tested. It is worth highlighting that for all the considered cases, the heuristic could find feasible solutions for all the 540 problem instances while the solver could do it for a maximum of 85% of the instances (457 out of 540 for $\epsilon^r = \epsilon^d = 0.8$). These results show the advantages of the heuristic when compared to a general-purpose optimization solver.

Additionally, Table 5.4 shows the results of the heuristic for the CRC case. We tested it under different uncertainty levels and for a value of the fixed contracting cost (f) of 5,000. In this case, the heuristic is also able to find feasible solutions for all the 540 instances of each combination of the uncertainty levels. The results show that the heuristic finds reasonably good feasible solutions for the CRC case using a small fraction of the CPU time spent by the solver.

Uncertainty level		CPLEX				PH		
ϵ^r	ϵ^d	#F	#O	Opt gap (%)	Time (secs)	Dif UB (%)	Dif LB (%)	Time (secs)
0.2	0.2	348	194	5.54	3,832.04	-2.74	3.22	615.41
0.4	0.4	397	296	3.24	2,653.74	-1.27	2.18	621.48
0.6	0.6	430	334	1.56	2,295.26	0.67	2.34	619.00
0.8	0.8	457	356	0.65	2,078.50	1.92	2.62	620.79
0.6	0.0	404	288	1.67	2,771.02	-0.35	1.38	618.05
0.0	0.6	368	261	3.64	3,009.63	-1.82	2.09	619.75

Table 5.3: Results with the PH-based heuristic for the lost sales case under different uncertainty levels

Uncertainty level		CPLEX				PH		
ϵ^r	ϵ^d	#F	#O	Opt gap (%)	Time (secs)	Dif UB (%)	Dif LB (%)	Time (secs)
0.2	0.2	375	257	4.85	3,146.52	-2.02	3.34	641.12
0.4	0.4	399	313	2.00	2,338.07	1.07	3.24	641.21
0.6	0.6	417	324	2.05	2,282.46	1.20	3.57	638.25
0.8	0.8	452	358	0.86	2,075.85	3.80	4.74	638.73
0.6	0.0	406	290	2.69	2,800.41	-0.64	2.22	640.63
0.0	0.6	393	289	4.31	2,714.34	2.20	7.28	632.92

Table 5.4: Results with the PH-based heuristic for the CRC case under different uncertainty levels

From all these experiments it was possible to observe that the solutions provided by the heuristic also present the protection mechanisms against the uncertainties of the optimal solutions, as described in the previous sections. In particular, for the lost sales case, the solutions tend to perform more visits and the delivery sizes tend to decrease when we increase the uncertainty levels. For the CRC case, the protection mechanism also includes increasing the number if contracts in the first stage when the uncertainty levels increase. These results highlight the advantages of the heuristic since, in addition to providing high-quality solutions in a small fraction of the CPU time used by the solver, its solutions are robust in terms of behavior against the problem uncertainties. It is worth mentioning that for the sake of brevity we do not display the results of the heuristic algorithm in their full extent.

5.6 Final remarks

In this chapter, we addressed an inventory routing problem under both stochastic product supply and customer demands. We introduced a two-stage stochastic programming formulation considering recourse mechanisms such as lost sales, backlogging and an additional source for the product in a capacity reservation contract setting. We also proposed a progressive hedging-based heuristic algorithm. We have provided several managerial insights regarding the behavior of the optimal solutions under different configurations for the uncertainty levels and costs of the system. Furthermore, the results with the heuristic algorithm showed that it provides high-quality solutions within reasonable running times for instances with a large number of scenarios.

Chapter 6

Conclusions

6.1 Concluding remarks

In the previous chapters, we have studied several variants of the IRP and presented different formulations and methods to solve them. By investigating different practical constraints for the IRP and providing tailored solution methods for the addressed variants, this thesis addresses problems arising in several contexts and shows the adaptability of the basic IRP and how it can be used as a basis to study richer practical IRPs. Furthermore, the competitive results obtained in this study demonstrate the ability of operations research to contribute for decision making in the supply chain context. Thus, in this section we provide a short summary of the main findings and contributions reported in each chapter of this thesis. The final section of this chapter concludes it by identifying possible directions for future research.

In Chapter 2, we presented a detailed description of the basic variant of the IRP. A mathematical formulation with two different objective functions was presented. The formulation was presented using an exponentially large number of subtour elimination constraints, as well as in a compact form. The former type of formulation provides stronger bounds than their respective compact counterparts but typically requires specialized separation procedures implemented within branch-and-cut schemes. The compact version has the advantage of being easily implementable using general-purpose optimization softwares although their bounds may be considerable weaker. Though relatively simple, this variant can be used as a basis to model richer practical variants of the problem given that no restricting assumption regarding the problem structure is included when defining it. Therefore, designing effective solutions methods for this variant becomes a relevant issue, particularly for the development of tools for decision-making in practice.

Building on this, in Chapter 3 we presented two metaheuristic algorithms based on iterated local search (ILS) and simulated annealing (SA) to solve the basic variant of the IRP, respectively. ILS is a metaheuristic that iterates between a local search heuristic and a perturbation algorithm, which guides the search over the space of local optimal solutions, whereas SA can probabilistically accept solutions that temporarily produce degradations in the current incumbent solution to avoid getting trapped in local optimal solutions. The metaheuristics were developed to solve the problem considering the classical total cost minimization objective function as well as the logistic ratio as alternative objective function. The results with the standard cost minimization showed that the presented metaheuristics can offer different advantages according to the instance characteristic, as none of them dominated the other in the whole set of benchmark instances tested. In particular, SA found better results, on average, for the instances with shorter planning horizons while ILS found slightly better results for the instances with a longer planning horizons.

The results minimizing the logistic ratio showed that SA outperformed ILS in all sets of instances used to test the algorithms, considering the average results, although both metaheuristics were able to find optimal or near optimal solutions for most of the analyzed instances. These results demonstrate the ability of the developed algorithms to also address this objective function. In addition, it was observed that the solutions with the logistic ratio as an objective function have an average cost that is more than 12% higher with respect to the solutions minimizing the total cost. However, an average reduction of 15% in the logistic ratio was reached when compared to the classical objective function. Lower logistic ratios are obtained for instances with fewer vehicles, as the travel costs tend to increase with the number of vehicles. All the experiments were based on 1098 problem instances from the literature, and the ILS and SA algorithms found, respectively, 289 and 283 solutions with objective values better than the best known solutions in the literature (for the first objective function).

In Chapter 4 we addressed a practical variant of the IRP considering product perishability. This feature gains significance in the supply chain context given that in many industries, raw materials, as well as intermediate and final products, are often perishable. Moreover, perishability may appear in more than one activity throughout the supply chain, which gives perishability a substantial significance in many practical settings, particularly in agri-food supply chains. In the variant studied in this thesis, the product was assumed to have a fixed shelf-life during which it is usable and after which it must be discarded. Age-dependent revenues and inventory holding costs were also considered. This type of perishability modeling is in line with the classification framework proposed by Amorim et al. (2013) for production and distribution planning. The authors state that perishability can be classified into three types: (*i*) associated to the physical deterioration of the products with time; (*ii*) related to the perceived value of the product for the customers; and (*iii*) associated to regulations that directly influence the occurrence of the spoilage event. Thus, the studied variant can be used as a basis to model several applications involving these three types of perishability, as it only assumes that the product has a fixed lifetime and no restricting assumption is made on the age-dependent revenue and holding cost values.

For this variant, we first introduced and compared four mathematical formulations, two with a vehicle index and two without a vehicle index. These formulations were solved using tailored branch-and-cut algorithms. The results show that the formulations without a vehicle index provide a considerably larger number of feasible and optimal solutions within the two-hours time limit, when compared to the other formulations, in addition to a significant speed-up for instances solved to optimality within the time limit by all the formulations. However, since the formulations without a vehicle index can only be used for the cases in which the vehicle fleet is homogeneous, it is worth mentioning that the proposed formulations with a vehicle index also provided a slightly superior performance when compared to a standard vehicle flow-based formulation.

Additionally, we developed a hybrid heuristic method based on the combination of an ILS metaheuristic and two mathematical programming components. This method was able to provide high-quality solutions within relatively short running times on small- and medium-sized problem instances. Specifically, solutions with an average optimality gap of 1.87% were obtained within seven seconds, on average. When applied to larger instances, the method provided good feasible solutions within reasonable running times. On average, the hybrid method found (within three minutes) solutions with objective values 1.73% better than the best solutions found by all the formulations for these instances. We also adapted the proposed hybrid heuristic to solve the basic variant of the IRP. The results using standard instances show that our heuristic is also able to find good quality solutions for this problem when compared to the state-of-the-art methods from the literature.

Finally, in Chapter 5 we shifted to a stochastic variant of the IRP. We considered the basic variant of the IRP under the consideration that both the product supply and the customer demands are uncertain. In the supply chain management context, taking uncertainty into account becomes relevant given that critical input data that are required for effective operations planning often are not known with certainty when the plan is made. This directly impacts the quality of the decisions as using inaccurate information can lead to poor performance in practice. Thus, we proposed a two-stage stochastic programming formulation for this problem, where routing decisions are made in the first stage, while delivery quantities, inventory levels and specific recourse actions are determined in the second stage. We considered different recourse mechanisms such as lost sales and backlogging as well as an additional source for the product in a capacity reservation contract setting. The objective was to minimize the first-stage cost plus the total expected inventory and recourse cost incurred in the second stage.

Experiments with the proposed model allowed us to provide managerial insights regarding the response mechanisms of the optimal solutions under different uncertainty levels of the random variables and different cost configurations. In particular, we showed that under the presence of high uncertainty levels the optimal protection mechanism consists of planning more visits with reduced delivery sizes, when compared to a scenario with low uncertainty. This applies for all the considered recourse actions. In the capacity reservation contract setting, this protection mechanism also includes increasing the number of contracts reserved in advance. In addition, it was possible to observe the trade-off between the reservation cost paid upfront and the purchasing cost resulting from the supplementary recourse of this setting.

We also proposed a heuristic solution method for the stochastic IRP. The method is based on the progressive hedging algorithm. The results showed that on average the heuristic was able to find reasonably high-quality feasible solutions within reasonable running times for instances with a large number of scenarios when compared to a general-purpose optimization software. Specifically, for an uncertainty level of 60% for both product supply and customer demands, the heuristic could find solutions with an average relative difference of less than 1% to those provided by the solver. These solutions have an average optimality gap of 2.34% and were obtained in an average CPU time of 619 seconds, which represents 27% of the time spent by the solver. It is worth highlighting that the heuristic was able to find feasible solutions for all the instances considered while the solver could find feasible solutions for only 80% of the instances.

6.2 Research opportunities

Building on the results of this thesis, in this section we discuss several ideas for potential future research. We include suggestions to extend the current research as well as ways to overcome some of the limitations of the studies presented in this thesis.

First, hybrid solution methods could be developed from the metaheuristic algorithms presented in Chapter 3. In particular, combining these metaheuristics (or part of them) with methods based on column generation has the potential to lead to effective hybrid methods, as successfully applied in the vehicle routing problem context (Alvarez and Munari, 2017; Alvarenga et al., 2007; Danna and Le Pape, 2005). The metaheuristic(s) could be used as column generator(s) while an optimization software would combine the generated routes using a column-based formulation. Alternatively, the metaheuristics could be included into a column generation-based method as a solution generation and improvement component. These principles apply for all the studied variants of the problem. Notice that an effective hybrid method based on the ILS metaheuristic of Chapter 3 and two mathematical programming components was proposed in Chapter 4 for the variant with perishable products.

Another interesting perspective for future research is the development of exact methods based on branch-and-price or column generation. The method could be developed on top of the interior point branch-price-and-cut framework proposed by Munari and Gondzio (2013), which uses the primal-dual interior point method available on the high order primal dual method solver (Gondzio, 1995). This framework relies on an interior point algorithm to solve the restricted master problems, which generally reduces the computational times related to solve large restricted master problems, in addition to simultaneously cope with column and cut stabilization (Munari and Gondzio, 2013; Gondzio et al., 2016). This might lead to advantages with respect to current exact methods based on branch-and-price algorithms available in the literature, which rely on simplex algorithms to solve the linear programs. It is worth pointing out that the performance of the exact methods proposed so far in the literature for the basic variant of the IRP (based on branch-and-cut and branch-and-price) worsens considerably when increasing the length of the planning horizon of the problem (Desaulniers et al., 2016; Archetti et al., 2017b; Coelho and Laporte, 2013b; Avella et al., 2018). Additionally, effective cutting plane algorithms recently developed by Desaulniers et al. (2016) and Avella et al. (2018) for the basic variant of the IRP could also be incorporated to boost the performance of the developed algorithm.

Regarding the research on perishability issues in the IRP context, we suggest three different lines for future research emerging from the study developed in this thesis. First, addressing extensions of the problem richer than those addressed in Chapter 4. For instance, cases with multiple perishable products as well as with multiple sources of the product could be considered with the aim of making the addressed problem as realistic as possible. Moreover, priority rules (e.g., freshest first or oldest first) for delivery and consumption variables could be useful to solve particular cases of the problem. As a second line of research, we suggest considering production as a decision within the problem, which would lead to a production routing problem with perishability considerations. Finally, future research could consider data uncertainty in the problem together with perishability-related parameters, particularly in the maximum age of the product.

Finally, future research on the stochastic IRP context could focus on considering other recourse as well as working on a multi-stage decision framework. In this sense, considering uncertainties in a multi-stage setting could be more accurate than a two-stage approach given the multi-period nature of the IRP. Additionally, developing effective exact methods for the stochastic programs presented in Chapter 5 arises as a natural line of research. In particular, methods based on Benders decomposition, exploiting the complete recourse nature of these programs, could be a promising approach for dealing with this problem.

Bibliography

Abdel-Malek, L. L. and Ziegler, H. (1988). Age dependent perishability in two-echelon serial inventory systems. *Computers & Operations Research*, 15(3):227–238.

Abdelmaguid, T. F., Dessouky, M. M., and Ordóñez, F. (2009). Heuristic approaches for the inventory-routing problem with backlogging. *Computers & Industrial Engineering*, 56(4):1519–1534.

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

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014b). Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48(1):20–45.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015a). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015b). The production routing problem: A review of formulations and solution algorithms. *Computers & Operations Research*, 55:141–152.

Al-Khayyal, F. and Hwang, S.-J. (2007). Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, part I: Applications and model. *European Journal of Operational Research*, 176(1):106–130.

Alvarenga, G., Mateus, G., and De, Tomi, G. (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34(6):1561–1584.

Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., and Morabito, R. (2020). Formulations, branch-and-cut and a hybrid heuristic algorithm for an inventory routing problem with perishable products. *European Journal of Operational Research*, 283(2):511–529.

Alvarez, A. and Munari, P. (2017). An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. *Computers & Operations Research*, 83:1–12.

Alvarez, A., Munari, P., and Morabito, R. (2018). Iterated local search and simulated annealing algorithms for the inventory routing problem. *International Transactions in Operational Research*, 25:1785–1809.

Amorim, P., Meyr, H., Almeder, C., and Almada-Lobo, B. (2013). Managing perishability in production-distribution planning: A discussion and review. *Flexible Services and Manufacturing Journal*, 25(3):389–413.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., and Løkketangen, A. (2010). Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536.

Applegate, D., Bixby, R., Chvátal, V., and Cook, W. (2018). Concorde TSP solver. `http://www.math.uwaterloo.ca/tsp/concorde.html`. Accessed: 2018-07-20.

Archetti, C., Bertazzi, L., Hertz, A., and Speranza, M. G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1):101–116.

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

Archetti, C., Bianchessi, N., Irnich, S., and Speranza, M. G. (2014). Formulations for an inventory routing problem. *International Transactions in Operational Research*, 21(3):353–374.

Archetti, C., Boland, N., and Speranza, M. G. (2017a). A matheuristic for the multivehicle inventory routing problem. *INFORMS Journal on Computing*, 29(3):377–387.

Archetti, C., Coelho, L. C., and Speranza, M. G. (2019). An exact algorithm for the inventory routing problem with logistic ratio. *Transportation Research Part E: Logistics and Transportation Review*, 131:96–107.

Archetti, C., Desaulniers, G., and Speranza, M. G. (2017b). Minimizing the logistic ratio in the inventory routing problem. *EURO Journal on Transportation and Logistics*, 6(4):289–306.

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

Avella, P., Boccia, M., and Wolsey, L. A. (2018). Single-period cutting planes for inventory routing problems. *Transportation Science*, 52(3):497–508.

Azadeh, A., Elahi, S., Farahani, M. H., and Nasirian, B. (2017). A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers & Industrial Engineering*, 104:124–133.

Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., Mack, R. G., and Prutzman, P. J. (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23.

Benoist, T., Gardi, F., Jeanjean, A., and Estellon, B. (2011). Randomized local search for real-life inventory routing. *Transportation Science*, 45(3):381–398.

Bertazzi, L., Bosco, A., Guerriero, F., and Laganà, D. (2013). A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies*, 27:89–107.

Bertazzi, L., Bosco, A., and Laganà, D. (2015). Managing stochastic demand in an inventory routing problem with transportation procurement. *Omega*, 56:112–121.

Bertrand, J. W. M. and Fransoo, J. C. (2002). Operations management research methodologies using quantitative modeling. *International Journal of Operations & Production Management*, 22(2):241–264.

Bertsekas, D. P. (1982). *Constrained optimization and Lagrange multiplier methods.* Athena Scientific.

Birge, J. R. and Louveaux, F. (2011). The value of information and the stochastic solution. In *Introduction to Stochastic Programming*, pages 163–177. Springer.

Bräysy, O. and Gendreau, M. (2005). Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, 39:104–118.

Burke, G. J., Carrillo, J. E., and Vakharia, A. J. (2009). Sourcing decisions with stochastic supplier reliability and stochastic demand. *Production and Operations Management*, 18(4):475–484.

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

Chen, Z. (2018). Optimization of production inventory with pricing and promotion effort for a single-vendor multi-buyer system of perishable products. *International Journal of Production Economics*, 203:333–349.

Chitsaz, M., Cordeau, J.-F., and Jans, R. (2019). A unified decomposition matheuristic for assembly, production, and inventory routing. *INFORMS Journal on Computing*, 31(1):134–152.

Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2007). Maritime transportation. In Barnhart, C. and Laporte, G., editors, *Handbooks in Operations Research and Management Science*, pages 189–284. Elsevier.

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

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

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

Coelho, L. and Laporte, G. (2013a). A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research*, 51(23):7156–7169.

Coelho, L. and Laporte, G. (2013b). The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, 40(2):558–565.

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

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

Cordeau, J.-F., Laganà, D., Musmanno, R., and Vocaturo, F. (2015). A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, 55(2):153–166.

Crainic, T., Fu, X., Gendreau, M., Rei, W., and Wallace, S. (2011). Progressive hedging-based metaheuristics for stochastic network design. *Networks*, 58(2):114–124.

Crama, Y., Rezaei, M., Savelsbergh, M., and Van Woensel, T. (2018). Stochastic inventory routing for perishable products. *Transportation Science*, 52(3):526–546.

Danna, E. and Le Pape, C. (2005). Branch-and-price heuristics: A case study on the vehicle routing problem with time windows. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, pages 99–129. Springer.

Deng, Y. and Bard, J. F. (2011). A reactive GRASP with path relinking for capacitated clustering. *Journal of Heuristics*, 17(2):119–152.

Desaulniers, G., Lessard, F., and Hadjar, A. (2008). Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3):387–404.

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

Diabat, A., Abdallah, T., and Le, T. (2016). A hybrid tabu search based heuristic for the periodic distribution inventory problem with perishable goods. *Annals of Operations Research*, 242(2):373–398.

Díaz-Madroñero, M., Peidro, D., and Mula, J. (2015). A review of tactical optimization models for integrated production and transport routing planning decisions. *Computers & Industrial Engineering*, 88:518–535.

Dillon, M., Oliveira, F., and Abbasi, B. (2017). A two-stage stochastic programming model for inventory management in the blood supply chain. *International Journal of Production Economics*, 187:27–41.

Dror, M. and Ball, M. (1987). Inventory/routing: Reduction from an annual to a short-period problem. *Naval Research Logistics*, 34:891–905.

Dror, M., Ball, M., and Golden, B. (1985). A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research*, 4(1):1–23.

Engineer, F. G., Furman, K. C., Nemhauser, G., Savelsbergh, M., and Song, J.-H. (2012). A branch-price-and-cut algorithm for single-product maritime inventory routing. *Operations Research*, 60(1):106–122.

Federgruen, A., Prastacos, G., and Zipkin, P. (1986). An allocation and distribution model for perishable products. *Operations Research*, 34(1):75–82.

Federgruen, A. and Zipkin, P. (1984). A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019–1037.

Galvão, R. D., Chiyoshi, F. Y., and Morabito, R. (2005). Towards unified formulations and extensions of two classical probabilistic location models. *Computers & Operations Research*, 32(1):15–33.

Gondzio, J. (1995). HOPDM (version 2.12) - a fast LP solver based on a primal-dual interior point method. *European Journal of Operational Research*, 85(1):221–225.

Gondzio, J., González-Brevis, P., and Munari, P. (2016). Large-scale optimization with the primal-dual column generation method. *Mathematical Programming Computation*, 8(1):47–82.

Gruler, A., Panadero, J., de Armas, J., Moreno Pérez, J. A., and Juan, A. A. (2018). Combining variable neighborhood search with simulation for the inventory routing problem with stochastic demands and stock-outs. *Computers & Industrial Engineering*, 123:278–288.

Gruson, M., Cordeau, J.-F., and Jans, R. (2018). The impact of service level constraints in deterministic lot sizing with backlogging. *Omega*, 79:91–103.

Gurobi Optimization, Inc. (2015). Gurobi optimizer reference manual.

Haugen, K., Løkketangen, A., and Woodruff, D. L. (2001). Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. *European Journal of Operational Research*, 132(1):116–122.

Hemmelmayr, V., Doerner, K., Hartl, R., and Savelsbergh, M. (2009). Delivery strategies for blood products supplies. *OR Spectrum*, 31(4):707–725.

Huang, S. H. and Lin, P. C. (2010). A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):598–611.

Hutter, F., Hoos, H. H., Leyton-Brown, K., and Stützle, T. (2009). ParamILS: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306.

Hvattum, L. M., Løkketangen, A., and Laporte, G. (2009). Scenario tree-based heuristics for stochastic inventory-routing problems. *INFORMS Journal on Computing*, 21(2):268–285.

IBM ILOG CPLEX (2009). User's manual for CPLEX. *International Business Machines Corporation.*

Jaillet, P., Bard, J. F., Huang, L., and Dror, M. (2002). Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science*, 36(3):292–300.

Jans, R. (2009). Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. *INFORMS Journal on Computing*, 21(1):123–136.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kleindorfer, P. R. and Wu, D. J. (2003). Integrating long- and short-term contracting via business-to-business exchanges for capital-intensive industries. *Management Science*, 49(11):1597–1615.

Kleywegt, A. J., Nori, V. S., and Savelsbergh, M. (2004). Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science*, 38(1):42–70.

Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2014). Vehicle routing problems in which consistency considerations are important: A survey. *Networks*, 64(3):192–213.

Krarup, J. and Bilde, O. (1977). Plant location, set covering and economic lot size: An 0 (mn)-algorithm for structured problems. In Collatz, L., Meinardus, G., and Wetterling, W., editors, *Numerische Methoden bei Optimierungsaufgaben Band 3: Optimierung bei graphentheoretischen und ganzzahligen Problemen*, pages 155–180. Birkhäuser Basel.

Lamghari, A. and Dimitrakopoulos, R. (2016). Progressive hedging applied as a metaheuristic to schedule production in open-pit mines accounting for reserve uncertainty. *European Journal of Operational Research*, 253(3):843–855.

Larrain, H., Coelho, L. C., and Cataldo, A. (2017). A variable MIP neighborhood descent algorithm for managing inventory and distribution of cash in automated teller machines. *Computers & Operations Research*, 85:22–31.

Le, T., Diabat, A., Richard, J.-P., and Yih, Y. (2013). A column generation-based heuristic algorithm for an inventory routing problem with perishable goods. *Optimization Letters*, 7(7):1481–1502.

Løkketangen, A. and Woodruff, D. L. (1996). Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming. *Journal of Heuristics*, 2(2):111–128.

Lourenço, H., Martin, O., and Stützle, T. (2003). Iterated local search. In Glover, F. and Kochenberger, G., editors, *Handbook of Metaheuristics*, pages 321–353. Springer.

Luenberger, D. G. and Ye, Y. (2016). Duality and dual methods. In *Linear and Nonlinear Programming*, pages 429–465. Springer.

Lysgaard, J., Letchford, A., and Eglese, R. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445.

Markov, I., Bierlaire, M., Cordeau, J.-F., Maknoon, Y., and Varone, S. (2018). A unified framework for rich routing problems with stochastic demands. *Transportation Research Part B: Methodological*, 114:213–240.

Markov, I., Bierlaire, M., Cordeau, J.-F., Maknoon, Y., and Varone, S. (2020). Waste collection inventory routing with non-stationary stochastic demands. *Computers & Operations Research*, 113:104798.

Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7(4):326–329.

Miranda, P. L., Cordeau, J.-F., Ferreira, D., Jans, R., and Morabito, R. (2018). A decomposition heuristic for a rich production routing problem. *Computers & Operations Research*, 98:211–230.

Mirzaei, S. and Seifi, A. (2015). Considering lost sale in inventory routing problems for perishable goods. *Computers & Industrial Engineering*, 87:213–227.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24:1097–1100.

Morabito, R. and Pureza, V. (2012). Modelagem e simulação. In Cauchick, P., editor, *Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações*, pages 169–198. Elsevier, 2nd edition.

Moreno, A., Alem, D., Ferreira, D., and Clark, A. (2018). An effective two-stage stochastic multi-trip location-transportation model with social concerns in relief supply chains. *European Journal of Operational Research*, 269(3):1050–1071.

Munari, P. and Gondzio, J. (2013). Using the primal-dual interior point algorithm within the branch-price-and-cut method. *Computers & Operations Research*, 40(8):2026–2036.

Neves-Moreira, F., Almada-Lobo, B., Cordeau, J.-F., Guimarães, L., and Jans, R. (2019). Solving a large multi-product production-routing problem with delivery time windows. *Omega*, 86:154–172.

Nikzad, E., Bashiri, M., and Oliveira, F. (2019). Two-stage stochastic programming approach for the medical drug inventory routing problem under uncertainty. *Computers & Industrial Engineering*, 128:358–370.

Nolz, P. C., Absi, N., and Feillet, D. (2014). A stochastic inventory routing problem for infectious medical waste collection. *Networks*, 63(1):82–95.

Öncan, T., Altınel, I. K., and Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3):637–654.

Pecin, D., Pessoa, A., Poggi, M., and Uchoa, E. (2014). Improved branch-cut-and-price for capacitated vehicle routing. In Lee, J. and Vygen, J., editors, *Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, pages 393–403. Springer.

Popović, D., Vidović, M., and Radivojević, G. (2012). Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. *Expert Systems with Applications*, 39(18):13390–13398.

Qiu, Y., Qiao, J., and Pardalos, P. M. (2019). Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory. *Omega*, 82:193–204.

Rockafellar, R. T. and Wets, R. J.-B. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147.

Rohmer, S., Claassen, G., and Laporte, G. (2019). A two-echelon inventory routing problem for perishable products. *Computers & Operations Research*, 107:156–172.

Salavati-Khoshghalb, M., Gendreau, M., Jabali, O., and Rei, W. (2019). An exact algorithm to solve the vehicle routing problem with stochastic demands under an optimal restocking policy. *European Journal of Operational Research*, 273(1):175–189.

Santos, E., Ochi, L., Simonetti, L., and González, P. (2016). A hybrid heuristic based on iterated local search for multivehicle inventory routing problem. *Electronic Notes in Discrete Mathematics*, 52:197–204.

Savelsbergh, M. and Song, J.-H. (2007). Inventory routing with continuous moves. *Computers & Operations Research*, 34(6):1744–1763.

Savelsbergh, M. and Song, J.-H. (2008). An optimization algorithm for the inventory routing problem with continuous moves. *Computers & Operations Research*, 35(7):2266–2282.

Schmid, V., Doerner, K. F., and Laporte, G. (2013). Rich routing problems arising in supply chain management. *European Journal of Operational Research*, 224(3):435–448.

Serel, D. A. (2007). Capacity reservation under supply uncertainty. *Computers & Operations Research*, 34(4):1192–1220.

Serel, D. A., Dada, M., and Moskowitz, H. (2001). Sourcing decisions with capacity reservation contracts. *European Journal of Operational Research*, 131(3):635–648.

Shaabani, H. and Kamalabadi, I. N. (2016). An efficient population-based simulated annealing algorithm for the multi-product multi-retailer perishable inventory routing problem. *Computers & Industrial Engineering*, 99:189–201.

Shiguemoto, A. L. and Armentano, V. A. (2010). A tabu search procedure for coordinating production, inventory and distribution routing problems. *International Transactions in Operational Research*, 17(2):179–195.

Singh, T., Arbogast, J. E., and Neagu, N. (2015). An incremental approach using local-search heuristic for inventory routing problem in industrial gases. *Computers & Chemical Engineering*, 80:199–210.

Slack, N., Chambers, S., and Johnston, R. (2010). *Operations management*. Pearson education.

Solyalı, O. and 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.

Solyalı, O. and Süral, H. (2012). The one-warehouse multi-retailer problem: reformulation, classification, and computational results. *Annals of Operations Research*, 196(1):517–541.

Song, J.-H. and Furman, K. C. (2013). A maritime inventory routing problem: Practical approach. *Computers & Operations Research*, 40(3):657–665.

Soysal, M., Bloemhof-Ruwaard, J., Haijema, R., and 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.

Soysal, M., Bloemhof-Ruwaard, J., Haijema, R., and Van Der Vorst, J. (2018). Modeling a green inventory routing problem for perishable products with horizontal collaboration. *Computers & Operations Research*, 89:168–182.

Stanzani, A., Pureza, V., Morabito, R., Da Silva, B., Yamashita, D., and Ribas, P. (2018). Optimizing multiship routing and scheduling with constraints on inventory levels in a Brazilian oil company. *International Transactions in Operational Research*, 25(4):1163–1198.

Subramanian, A., Drummond, L. M., Bentes, C., Ochi, L. S., and Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911.

Tempelmeier, H. (2013). Stochastic lot sizing problems. In Smith, J. M. and Tan, B., editors, *Handbook of Stochastic Models and Analysis of Manufacturing System Operations*, pages 313–344. Springer.

Toth, P. and Vigo, D. (2002). An overview of vehicle routing problems. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem*, pages 1–26. SIAM.

Vladimirou, H. and Zenios, S. A. (1997). Stochastic programming and robust optimization. In Gal, T. and Greenberg, H. J., editors, *Advances in Sensitivity Analysis and Parametic Programming*, pages 395–447. Springer.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Yu, M. and Nagurney, A. (2013). Competitive food supply chain networks with application to fresh produce. *European Journal of Operational Research*, 224(2):273–282.

Yu, V. F. and Lin, S. W. (2014). Multi-start simulated annealing heuristic for the location routing problem with simultaneous pickup and delivery. *Applied Soft Computing Journal*, 24:284–290.

Yu, Y., Chu, C., Chen, H., and Chu, F. (2012). Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research*, 197(1):135–158.

Zeballos, L. J., Méndez, C. A., Barbosa-Povoa, A. P., and Novais, A. Q. (2014). Multi-period design and planning of closed-loop supply chains with uncertain supply and demand. *Computers and Chemical Engineering*, 66:151–164.