HEC MONTRÉAL
AFFILIÉE À L'UNIVERSITÉ DE MONTRÉAL

Flexibility and Consistency in Inventory-Routing

par
Leandro Callegari Coelho

Thèse présentée à la Faculté des études supérieures et postdoctorales
en vue de l'obtention du grade de Ph.D. en administration

Août 2012

HEC MONTRÉAL
AFFILIÉE À L'UNIVERSITÉ DE MONTRÉAL

Cette thèse intitulée:

Flexibility and Consistency in Inventory-Routing

présentée par

Leandro Callegari Coelho

a été evaluée par un jury composé des personnes suivantes:

Raf Jans
président-rapporteur

Jean-François Cordeau
co-directeur de recherche

Gilbert Laporte
co-directeur de recherche

Walter Rei
membre du jury

Karen Smilowitz
examinateur externe

Roch Ouellet
représentant du directeur

Résumé

Dans plusieurs contextes, la logistique permet d'atteindre des avantages concurrentiels et des économies de coûts. Pour certaines entreprises, la logistique représente même leur compétence de base (soit les prestataires logistiques). Dans ce contexte, le système de *réapprovisionnement géré par le fournisseur* (RGF) est l'un des systèmes les plus à jour permettant aux entreprises d'atteindre des performances supérieures. En vertu d'une stratégie de RGF, les décisions liées au réapprovisionnement et à la distribution sont centralisées au niveau du fournisseur, ce qui entraine une réduction globale des coûts logistiques. Afin de faire fonctionner un système RGF, un Problème de Tournées et d'Approvisionnement (PTA) doit être résolu, en optimisant simultanément la gestion des stocks et les tournées des véhicules sur plusieurs périodes. Notre but est d'introduire deux nouveaux concepts, appelés *flexibilité* et *régularité*, dans le cadre du PTA.

La *flexibilité* sera prise en compte en considérant la possibilité de partager des stocks entre différents sites, ce qui rend le concept de *transbordement* applicable dans le contexte du PTA. Elle est également utile pour réagir rapidement à des fluctuations de demande dans un environnement dynamique et stochastique. Les problèmes de transbordement sont typiquement caractérisés par des mouvements de marchandises entre des entités de même niveau, comme les clients. Ils permettent au système de partager les risques de rupture de stock et d'accroitre la flexibilité du décideur en augmentant le nombre de sources à partir desquelles les marchandises peuvent être transférées. Le Problème de Tournées et d'Approvisionnement avec Transbordement (PTAT) est ensuite présenté. Il s'agit d'un problème pour lequel le décideur a la possibilité de planifier les mouvements de transbordement afin de minimiser le coût total du système. Ce problème se pose, par exemple, lorsque la résolution d'un Problème de Tournées et d'Approvisionnement Stochastique (PTAS) est considérée dans le cadre d'un horizon glissant où l'on utilise les prévisions de demande au cours des prochaines périodes comme une approximation de la demande future qui est incertaine. Nous présentons une formulation qui permet des transbordements, soit par le fournisseur à ses clients ou entre les clients. Nous développons un algorithme de séparation et coupes capable de résoudre des instances de petite et moyenne tailles. Nous présentons également une heuristique de recherche adaptative à grand voisinage (RAGV) pour résoudre de plus grandes instances. Cette heuristique manipule les tournées des véhicules alors que le problème qui consiste à déterminer les quantités à livrer ainsi que les mouvements de transbordement est résolu par un algorithme de flot dans un réseau. Notre approche permet de résoudre quatre variantes du problème: le PTA et le PTAT, avec réapprovisionnement flexible jusqu'au niveau maximal et avec réapprovisionnement fixé au niveau maximal. Une évaluation exhaustive de la performance de notre heuristique est effectuée.

La *régularité* aidera à améliorer la qualité du service, bénéficiant à la fois au fournisseur et aux clients en leur offrant des services plus réguliers. L'intégration des caractéristiques de régularité dans le cadre du PTA est aussi proposée. Celles-ci peuvent être utilisées pour améliorer la qualité du service offert par des solutions du PTA, rendant l'environnement moins tumultueux tout en fournissant des opérations plus stables à la fois pour le fournisseur et pour les clients. Nous analysons ensuite les PTA multi-véhicules (PTAM). Alors que les solutions qu'ils produisent ont tendance à avantager à la fois le fournisseur et ses clients, résoudre les PTAM uniquement en fonction de considérations de coût peut causer des inconvénients aux deux parties. Ceux-ci sont liés à la taille de la flotte et à la charge des véhicules, à la fréquence des livraisons, ainsi qu'aux quantités livrées. L'utilisation de plusieurs véhicules dont la capacité est peu utilisée, des visites très fréquentes ou rares pour un même client, et des fluctuations élevées dans les quantités à livrer constituent quelques exemples de tels inconvénients. Afin d'atténuer ces problèmes, nous introduisons le concept de régularité dans les solutions du PTA, augmentant ainsi la qualité du service. Nous avons formulé le PTAM comme un programme linéaire en variables mixtes et avons proposé un algorithme de séparation et coupes ainsi qu'une matheuristique pour sa résolution. Cette heuristique applique un système de RAGV dans laquelle certains sous-problèmes sont résolus exactement. L'algorithme proposé génère des solutions offrant un bon compromis entre le coût et la qualité. L'effet des politiques de gestion de stocks, des décisions de routage et des tailles de livraison est analysé.

Finalement, nous étendons notre analyse à la version dynamique et stochastique du PTA (PTAS). Nous intégrons les notions de flexibilité et de régularité dans la modélisation et la résolution du problème et nous analysons l'impact de politiques différentes, dans un contexte dans lequel toutes les informations ne sont pas disponibles pour le décideur. Nos politiques de travail sont basées sur un schéma d'horizon glissant. Nous comparons les effets de transbordement permettant d'atténuer les ruptures de stock et d'envisager des estimations des besoins futurs dans le processus de décision. Nous étudions aussi l'impact de l'introduction des critères de régularité sur la qualité du service.

La thèse est structurée comme suit. Après un chapitre introductif et de motivation, nous présentons une revue de littérature sur les thèmes pertinents, suivie de trois chapitres sur le problème des tournées et d'approvisionnement avec transbordement, le problème de tournées et d'approvisionnement avec régularité et le problème de tournées et d'approvisionnement dans un contexte dynamique et stochastique. Des conclusions et pistes de recherche sont présentées dans le dernier chapitre.

Mots clés: Problème de tournées et d'approvisionnement; flexibilité; régularité; heuristique; RAGV; déterministe; stochastique.

Abstract

In many contexts, logistics is used to enable competitive advantages and cost savings. For some companies, logistics itself is its core competency (i.e., logistics providers). In this context, vendor-managed inventory (VMI) systems are one of the most up-to-date strategies allowing companies to reach a superior performance. Under a VMI strategy, the replenishment and distribution making process is centralized at the supplier's level, leading to an overall reduction of logistics costs. In order to operate a VMI system, an *Inventory-Routing Problem* (IRP) has to be solved, simultaneously making inventory management and routing decisions over several periods. Our purpose is to introduce two new concepts, called *flexibility* and *consistency*, within the context of the IRP.

Flexibility will be added through the possibility of sharing inventory among locations, making the concept of *transshipment* available within inventory-routing. It is also useful to react quickly to changes in the demand in a dynamic and stochastic environment. Transshipment problems are typically characterized by movements of goods among entities of the same level, such as customers. They allow the system to share stockout risks and to increase the flexibility of the decision maker by increasing the number of sources from which goods can be transferred. We then introduce the *Inventory-Routing Problem with Transshipment* (IRPT), a problem in which the decision maker has the option to plan transshipment movements so as to minimize the total system cost. This problem arises, for instance, when solving stochastic Inventory-Routing Problems (SIRP) in a rolling horizon framework where one uses demand forecasts for the next time periods as approximations of the unknown demand. We present a formulation that allows transshipments, either from the supplier to customers or between customers. We develop a branch-and-cut algorithm capable of solving small and medium size instances. We also propose an adaptive large neighborhood search heuristic to solve larger instances. This heuristic manipulates vehicle routes while the remaining problem of determining delivery quantities and transshipment moves is solved through a network flow algorithm. Our approach can solve four different variants of the problem: the IRP and the IRPT, under maximum level and order-up-to level policies. We perform an extensive assessment of the performance of our heuristic.

Consistency will help offer higher quality of service, benefiting both supplier and customers with more regular services. We also propose the inclusion of consistency features within the IRP framework. They can be used to improve the quality of service offered through the IRP solutions, making the environment less noisy and providing smoother operations, both to the supplier and to the customers. Later, we analyze the *multi-vehicle IRP* (MIRP). Whereas the solutions they yield tend to benefit both the vendor and customers, solving MIRPs solely based on cost considerations may lead to inconveniences to both parties. These are related to the fleet size and vehicle load, to the frequency of the deliveries, and to the quantities delivered. The use of many vehicles with very low capacity utilisation, very frequent or rare visits to the same customer, and ever changing delivery quantities are some examples of such inconveniences. In order to alleviate these problems, we introduce the concept of consistency in IRP solutions, thus increasing quality of service. We formulate the multi-vehicle IRP as a mixed integer linear program and we propose a branch-and-cut algorithm and a matheuristic for its solution. This heuristic applies an ALNS scheme in which some subproblems are solved exactly. The proposed algorithm generates solutions offering a good compromise between cost and quality. We analyze the effect of different inventory policies, routing decisions and delivery sizes.

Finally, we extend our analysis to the study of the dynamic and stochastic version of the problem (SIRP). We integrate the notions of flexibility and consistency to the modeling and solution of this problem and we evaluate the impact of different policies in a context in which not all information is available to the decision maker. Our policies are developed in the context of a rolling horizon scheme. We compare the effects of allowing transshipments to mitigate stockouts and to consider estimates of future demands in the decision making process. We also study the impact of applying consistency policies on quality of service.

The thesis is structured as follows. After an introductory and motivational chapter, we present the literature review of the related themes, followed by three chapters on the inventory-routing problem with transshipment, the consistent inventory-routing problem and the dynamic and stochastic inventory-routing. Conclusions and directions for future work are presented in the last chapter.

Keywords: Inventory-routing problem; flexibility; consistency; heuristic; ALNS; deterministic; stochastic.

Resumo

Em muitas situações a logística é uma ferramenta que aumenta a vantagem competitiva e diminui os custos. Para algumas empresas, a própria logística é a sua competência central (como para os provedores logísticos). Neste contexto, o sistema de estoque gerenciado pelo fornecedor (EGF) é uma das estratégias mais atuais que permite que as empresas obtenham um melhor desempenho. Sob uma estratégia de EGF os processos de reposição e distribuição são centralizados pelo fornecedor, gerando uma redução dos custos logísticos. Para a operação de um sistema EGF, um Problema de Estoques e Roteamento (PER) deve ser resolvido, otimizando simultaneamente um problema de gerenciamento de estoques e um problema de roteamento de veículos, por vários períodos. O objetivo desta tese é introduzir dois novos conceitos no contexto do PER, chamados de *flexibilidade* e *consistência*.

A *flexibilidade* será adicionada com a possibilidade de compartilhar estoque entre vários locais, tornando o conceito de *transbordo* disponível para o PER. Ela também é útil para possibilitar uma reação rápida às mudanças na demanda em um ambiente dinâmico e estocástico. Os problemas de transbordo são normalmente caracterizados pelo transporte de bens entre elos de mesmo nível, como os varejistas. Eles permitem que todo o sistema compartilhe os riscos de falta de estoque e aumentam a flexibilidade do tomador de decisões ao aumentar o número de fontes de onde os produtos podem ser transferidos. Em seguida, introduzimos o Problema de Estoque e Roteamento com Transbordo (PERT), problema este onde o tomador de decisões tem a opção de planejar transbordos para minimizar o custo de operação do sistema. Esta situação ocorre, por exemplo, quando Problemas de Estoque e Roteamento Estocásticos (PERE) são resolvidos em um ambiente de horizonte rolante, onde se utilizam previsões de demanda para os próximos períodos como aproximações de uma demanda futura desconhecida, tornando-o determinístico no curto prazo. Apresentamos uma formulação que permite transbordos, tanto a partir do fornecedor para os varejistas quanto entre os varejistas. Desenvolvemos um algoritmo de *branch-and-cut* capaz de resolver instâncias de tamanho pequeno e médio. Também propomos uma heurística de busca em grande vizinhança adaptativa (BGVA) para resolver instâncias maiores. Esta heurística manipula as rotas dos veículos enquanto a determinação das quantidades a serem entregues, bem como os movimentos de transbordo são resolvidos através de um algoritmo de fluxo em grafo. Nossa abordagem permite solucionar quatro variantes do problema: o PER e o PERT, obedecendo estratégias de ressuprimento flexível com nível máximo e de ressuprimentos fixo ao nível máximo. Nós efetuamos uma extensa avaliação do desempenho da heurística.

A *consistência* ajudará a oferecer uma melhor qualidade no serviço, beneficiando tanto o fornecedor quanto os clientes com serviços mais regulares. A inclusão de caracterésticas de consistência no contexto do PER também é proposta. Elas podem ser usadas para melhorar a qualidade do serviço oferecido pelas soluções do PER ao tornar o ambiente menos tumultuado, gerando operações mais estáveis, tanto para o fornecedor quanto para os clientes. Depois, analisamos o PER com múltiplos veículos (PERM). Apesar de as soluções geradas tenderem a beneficiar tanto distribuidores quanto varejistas, a resolução do PERM considerando apenas os custos pode gerar inconvenientes para as duas partes, relacionados ao tamanho da frota utilizada e a carga em cada veículo, à frequência das entregas e às quantidades entregues. O uso de vários veículos com baixa utilização da capacidade, visitas muito frequentes ou raras ao mesmo cliente, e quantidades de entrega em constante mudança são alguns exemplos de tais inconvenientes. Para contornar estes problemas, introduzimos o conceito de consistência nas soluções do PER, aumentando assim a qualidade do serviço oferecido. Formulamos o PERM como um programa linear inteiro misto e propomos um algoritmo de *branch-and-cut* e uma heurística baseada na formulação matemática para a sua solução, que utiliza um esquema de BGVA onde alguns subproblemas são resolvidos de maneira exata. O algoritmo proposto gera soluções que oferecem um bom equilíbrio entre custo e qualidade. Nós avaliamos os efeitos de diferentes estratégias de gerenciamento de estoque, decisões de roteamento e tamanho das entregas.

Finalmente, a análise é extendida para o estudo da versão dinâmica e estocástica do problema (PERE). A integração de noções de flexibilidade e de consistência é feita na modelagem e resolução do problema, avaliando o impacto de diversas estratégias de ressuprimento em um contexto onde nem toda a informação está disponível no momento da tomada de decisão. Nossas estratégias são baseadas em um esquema de horizonte rolante. Comparamos os efeitos de permitir transbordos para reduzir as faltas de estoque e de considerar previsões sobre a demanda futura no proceso de tomada de decisão. O impacto da aplicação das políticas de consistência na qualidade do serviço também são estudadas.

Esta tese está estruturada da seguinte maneira: após um capítulo introdutivo e motivacional, uma revisão da literatura dos temas relacionados é apresentada, seguida por três capítulos sobre o problema de estoques e roteamento com transbordo, o problema de estoques e roteamento com consistência, e o problema de estoques e roteamento dinâmico e estocástico. Conclusões e sugestões para trabalhos futuros são apresentadas no último capítulo.

Palavras-chave: Problema de estoque e roteamento; consistência; flexibilidade; heurística; BGVA; determinístico; estocástico.

Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

ALNS	Adaptive Large Neighborhood Search.
DSIRP	Dynamic and Stochastic Inventory-Routing Problem.
EOQ	Economic Order Quantity.
GRASP	Greedy Randomized Adaptive Search Procedure.
IRP	Inventory-Routing Problem.
IRPT	Inventory-Routing Problem with Transshipment.
OU	Order-up-to level.
MIP	Mixed-Integer Program.
MILP	Mixed-Integer Linear Program.
MIRP	Multi-vehicle Inventory-Routing Problem.
ML	Maximum level.
PVRP	Periodic Vehicle Routing Problem.
SIRP	Stochastic Inventory-Routing Problem.
TSP	Traveling Salesman Problem.
VMI	Vendor-managed inventory.
VRP	Vehicle Routing Problem.
VRPTW	Vehicle Routing Problem with Time Windows.

Acknowledgments

I will always feel fortunate and honored for having had the pleasure of working with my two advisors Jean-François Cordeau and Gilbert Laporte. Their support, availability, patience and guidance have pushed me to offer my best, often more than I thought I was capable of doing. I am grateful to Professor Walter Rei for his support and his participation in my thesis committee, and to Professor Raf Jans who chaired the jury. I also thank the external examiner, Professor Karen Smilowitz, for her careful reading of my thesis and for her helpful suggestions.

I am also thankful for the unique scientific environment available both at HEC Montréal and at the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and to their respective staff, always willing to help whenever I needed. My friends and colleagues also played an important role in the development of my learning and my research. I could not start citing names as the list would need pages, but I am thankful to the many people I came to know during this Ph.D. I also thank the RQCHP (Réseau québécois de calcul de haute performance) for its assistance in providing excellent computing facilities.

Finally, this list of acknowledgements would be incomplete without my deepest sincere appreciation for my parents, Ludmar and Márcia, my sister Cíntia and my girlfriend Vanessa. Only with their support when hard times arrived was I able to continue this journey. Their encouragement started even before I moved to Montréal and has only increased to this day.

I also aknowledge the financial support offered by the Canadian Natural Sciences and Engineering Research Council, by the Canada Research Chair in Distribution Management, by the Canada Research Chair in Logistics and Transportation and by HEC Montréal.

I must share with these individuals any merit found in this research. Any shortcomings remain, of course, my own.

Chapter 1

Introduction

Logistics is now widely recognized as a value adding center in organizations through product availability, consistency of deliveries, accuracy in inventory and demand management, and ease of placing orders. Vendor-managed inventory (VMI) is one of the most up-to-date examples of value added through logistics.

In VMI systems, the replenishment and distribution making process is centralized at the supplier's level, based on specific inventory and supply chain policies and constitutes a streamlined approach to inventory management. The application of this policy leads to an overall reduction of logistics costs (Lee and Seungjin, 2008) and is often described as a win-win situation: suppliers save on distribution and production costs since they can combine and coordinate demands and shipments for different customers, and customers gain by not allocating resources to controlling and managing inventories. The supplier then has to make three simultaneous decisions: 1) when to serve a given customer, 2) how much to deliver, and 3) how to combine customers into routes.

The drawback of VMI is that it requires the solution of a very difficult combinatorial optimization problem, called the Inventory-Routing Problem (IRP), itself a combination of two well-studied problems: inventory management and vehicle routing. According to Andersson et al. (2010) "no commercially available systems provide decision support for combined inventory management and routing". Scientific research on the IRP is relatively recent compared to that on other optimization problems, such as the Vehicle Routing Problem (VRP). Speranza and Ukovich (1994) note the existence of distinct extensive literature reviews on transportation and on inventory management problems, but relatively few studies exploit their integration. This is still true to this day. A quick search on the ABI/INFORM Global database shows over 580 scholarly publications on the VRP, but less than 70 on the IRP. Recent reviews on the IRP found fewer than a hundred papers addressing the combined

VRP-inventory management problem (Andersson et al., 2010; Cordeau et al., 2007).

Most previous research is on standard versions of the IRP with deterministic demand, single vehicle, single product, operating policy aimed at minimizing the combined inventory-distribution cost.

Our aim is to treat a broader version of the problem. More specifically we will focus on two aspects that have been mostly neglected in previous work. The first, called *flexibility* will be added through the possibility of sharing inventory among locations, thus allowing transshipments within inventory-routing. It is also useful to react quickly to demand changes in a dynamic and stochastic environment. The second aspect, called *consistency*, will help offer higher quality of service, benefiting both supplier and customers with more regular services.

Gaining flexibility through transshipments has already been studied in the context of inventory management. Under this policy, goods may be shipped to a customer, either directly from the supplier, or from another customer. This happens, for example, between stores belonging to the same chain which can ship merchandise to one another when unforeseen demand variations occur (Axsäter, 1990; Dada, 1992; Lee, 1987; Nonås and Jörnsten, 2005, 2007; Paterson et al., 2011). Transshipments have been studied within the context of inventory management since the seminal paper of Allen (1958). A good analysis is presented in Herer et al. (2002). To the best of our knowledge, transshipment has not yet been formally integrated within the context of inventory-routing. Planned transshipments can also be used to redistribute inventory among customers so as to reduce handling costs, as is the case in the retail industry (Paterson et al., 2011) and in companies that make use of external freight carriers for part of their distribution (Nonås and Jörnsten, 2007). Transshipments may be beneficial in a deterministic context in which no shortages occur because they may yield an overall reduced distribution and inventory holding cost. This is the case, for example, when vehicle capacity and storage limits at customer locations restrict the amounts that can be delivered to these customers at each time period. Deterministic subproblems also arise when solving stochastic inventory-routing problems in a rolling horizon framework where one uses demand forecasts for the next time periods as approximations of the unknown demand. This is the context in which our problem is defined. Mercer and Tao (1996) provide an example of an inventory-routing system used by the supermarket chain Tesco, in the United Kingdom, where deliveries are made from a factory to several warehouses, and lateral transshipments can take place between warehouses. Obviously, the addition of transshipment within inventory-routing adds a layer of complexity to an already difficult problem.

Our treatment of consistency will extend some concepts previously presented within the VRP framework. We will add regularity features to the IRP by considering not only cost, but also quality of the service. Given that companies need not only provide cost effective solutions to their customers, but also high quality service, consistent solutions can be partly achieved by incorporating quality of service features in IRP solutions, which should yield a competitive advantage. This can be accomplished, for example, through the application of workforce management policies (Barlett and Ghoshal, 2002; Smilowitz et al., 2012; Groër et al., 2009). Thus, one would expect that regularly assigning the same driver to customers will help create a bond that can benefit both parties. Drivers will gain an increased familiarity with the region and the customer sites assigned to them, and will thus develop a more personal rapport with the customers. Another example of consistency is the spacing of deliveries to customers. To ensure smoother operations, visits should ideally be spread out evenly over the planning horizon. This type of requirement is often modeled as constraints in the context of the periodic Vehicle Routing Problem (VRP) (Christofides and Beasley, 1984; Francis et al., 2008) but it has not yet been imposed in the IRP. Finally, the quantities delivered to customers can also be controlled in order to avoid large variations over time, which are negatively perceived by customers (Beamon, 1999). Such regularity features will make the time interval between visits, the quantities delivered and the vehicle utilisation more consistent, thus offering higher service quality. We propose limiting the use of many vehicles with very low loads, avoiding very frequent or rare visits to the same customer, and ensuring that delivery quantities do not fluctuate too much over time. We also evaluate the impact of ensuring more stable operations not only on the cost of the final solution, but also in terms of the solution process.

Finally, we extend our analysis to the study of the dynamic and stochastic version of the problem (DSIRP). In this version of the problem, demand is dynamically revealed over time but one can exploit its statistical distribution in the solution process. We integrate the notions of flexibility and consistency to the modeling and resolution of this problem and we evaluate the impact of different policies in a context in which not all information is available to the decision maker. Our policies are developed in the context of a rolling horizon scheme. We compare the effects of allowing transshipments to mitigate stockouts and to consider estimates of future demands in the decision making process. We also study the impact of applying consistency policies on quality of service.

As we can see, both flexibility and consistency have been presented in different contexts, but their integration within the IRP is first seen in this thesis.

The remainder of this thesis is organized as follows. In Chapter 2 we present a survey of the literature and review several variants of the IRP that have arisen since this problem was first introduced by Bell et al. (1983). These include the IRP with a single customer (Bertazzi and Speranza, 2002; Dror and Ball, 1987; Speranza and Ukovich, 1996), the IRP with multiple customers (Archetti et al., 2007; Bell et al., 1983; Chien et al., 1989; Kleywegt et al., 2002), the stochastic IRP (Kleywegt et al., 2002, 2004; Minkoff, 1993), the IRP with direct deliveries (Gallego and Simchi-Levi, 1990, 1994; Hall, 1992; Kleywegt et al., 2002; Mishra and Raghunathan, 2004), the multi-item IRP (Bausch et al., 1998; Qu et al., 1999; Sindhuchao et al., 2005; Speranza and Ukovich, 1994), and the IRP with heterogeneous fleet (Chien et al., 1989; Christiansen, 1999; Persson and Göthe-Lundgren, 2005). There are so many ways of modeling and solving IRPs that different authors rarely define the problem in exactly the same way. In addition, real-life problems combining vehicle routing and inventory management concerns are often dynamic or stochastic. We provide a classification of the different variants, models and algorithms. We will also review the relevant literature about transshipments, as we will integrate it within the IRP framework in Chapter 3.

Then, in Chapter 3 we introduce the concept of *transshipment* within inventory-routing. Under this problem, goods may be shipped not only from the supplier to customers, but also among customers. This occurs, for example, between stores belonging to a same chain (Nonås and Jörnsten, 2005, 2007) which can ship merchandise to one another if it is needed and economically interesting. From a practical point of view, the use of a transshipment option contributes to lead-time and cost reductions.

We then analyze several *consistency* features in Chapter 4. Specifically, we study how much the solution cost increases if one ensures consistency in the deliveries. This consistency can be related to the quantities delivered to customers, to the time interval between consecutive visits to the same customer or to the vehicle capacity utilization. In this chapter we also extend the benchmarks available to the single vehicle IRP by considering a multi-vehicle framework.

Chapter 5 is devoted to the study of the dynamic and stochastic IRP, a problem where information is not completely available at the moment when decisions are made. We combine the flexibility and consistency concepts introduced earlier to the dynamic and stochastic IRP. Flexibility is obviously important in a dynamic and stochastic environment, due to its nature: it is needed to adapt the delivery plans to the ever changing nature of the demand; transshipments can also act as a means to protect retailers against demand fluctuations. Consistency is also important because in a dynamic context, the customers' actions are likely to be more variable than in the deterministic case and the consistency measures studied in Chapter 4 may therefore act as a way to protect the retailers against variations in service delivery. We evaluate the impact of different replenishment policies, the availability of information and the use of transshipments to mitigate the effects of stockouts.

Finally, Chapter 6 summarizes the main contributions of this thesis and points to some potential research directions.

Chapter 2

Literature review

Chapter information

An article partly based on this chapter was submitted for publication in *Transportation Science*: L. C. Coelho, J.-F. Cordeau, G. Laporte. Thirty Years of Inventory-Routing. Technical Report, *CIRRELT-2012-52*, Montreal, 2012.

In this chapter we present a survey of the literature and review several variants of the IRP. We also review the relevant literature on transshipment problems arising in inventory management in so far as it can be relevant to the IRP. We also describe some consistency features that have been used in vehicle routing and could also be applied to the IRP.

2.1 Introduction

In the last decades we have witnessed significant changes in the role of logistics management and an increased attention to this area. From a cost center, logistics is now seen as a value adding center, through product availability, consistency of deliveries, more precise inventory and demand management, ease of placing orders among other elements of the logistics service. Vendor-Managed Inventory (VMI) provides an efficient mechanism for adding value through logistics. It is a streamlined approach to inventory management: it connects a vendor (or supplier) closely to the customers (or buyers), with the former making the replenishment decisions for products supplied to the latter, based on specific inventory and supply chain policies (Angulo et al., 2004; Simchi-Levi et al., 2005; Lee and Seungjin, 2008).

VMI is often described as a win-win situation: vendors save on distribution and production costs as they are able to combine and to coordinate demands and shipments for different customers; buyers save by not allocating efforts to controlling and managing inventories. The supplier has to make three simultaneous decisions:

1. when to serve a given customer;

2. how much to deliver to this customer, when he is served;

3. how to combine customers into routes.

The drawback of VMI is that it requires the solution of a very difficult mathematical problem, called the Inventory-Routing Problem (IRP), a combination of two well-studied problems: (1) inventory management and (2) vehicle routing.

Table 2.1 shows how the IRP variants can be classified, according to eight criteria, based on Andersson et al. (2010), but not limited to their classification, namely time, demand, structure, routing, inventory policy, inventory decisions, fleet composition and fleet size.

Table 2.1: Classification used for the inventory-routing problem

Criteria	**Possible options**		
Time	Finite	Inifinite	
Demand	Deterministic	Stochastic	Dynamic
Structure	One-to-one	One-to-many	Many-to-many
Routing	Direct	Multiple	Continuous
Inventory policy	Unconstrained	Order-up-to level	
Inventory decisions	Lost sales	Back-order	Non-negative
Fleet composition	Homogeneous	Heterogeneous	
Fleet size	Single	Multiple	Unconstrained

Source: Adapted from Andersson et al. (2010)

In Table 2.1, time refers to the horizon taken into account by the IRP model. It can either be a finite horizon or an infinite horizon planning period. Demand from the customers can be either deterministic, stochastic or dynamic, and this is one of the major criteria as it defines a great part of the problem. Also, the number of suppliers and customers may change, thus the structure can be one-to-one when there is only one supplier serving one customer, one-to-many in the most common case with one supplier and several customers, and the less studied case many-to-many with several suppliers and several customers. This last configuration will only be cited in this research since there are substantially fewer papers studying this case. Routing options can be direct shipping, when there is only one customer in a route, multiple in the case where there are several customers being served by one vehicle on the same route, and continuous in cases where there is no central depot, like in several maritime applications. Inventory policies define pre-established rules to replenish customers. The options found in the literature are either unconstrained policies or a fixed policy called the order-up-to level. Inventory decisions determine how inventory management is modeled. If the inventory is allowed to become negative, then back-ordering occurs and the corresponding demand will be served when new shipments are delivered. If there is no back-order, then extra demand is considered as lost sales, and in both cases there may be a penalty for the stockout. In deterministic contexts, one can also restrict the inventory to be non-negative. Finally, the last two criteria refer to fleet composition and size. The fleet can either be homogeneous or heterogeneous, and the number of vehicles available may be fixed at one, fixed at many, or unconstrained. Each one of the following mentioned papers lies within some of these categories, and they will be classified accordingly.

Specific versions of the IRP include, but are not limited to the IRP with single customer (Dror and Ball, 1987; Speranza and Ukovich, 1996; Bertazzi and Speranza, 2002; Solyalı and Süral, 2008), the IRP with multiple customers (Archetti et al., 2007; Bell et al., 1983; Chien et al., 1989; Kleywegt et al., 2002), the stochastic IRP (Minkoff, 1993; Kleywegt et al., 2002, 2004), the IRP with direct deliveries (Gallego and Simchi-Levi, 1990, 1994; Hall, 1992; Kleywegt et al., 2002; Mishra and Raghunathan, 2004; Bertazzi, 2008), the multi-item IRP (Bausch et al., 1998; Qu et al., 1999; Sindhuchao et al., 2005; Speranza and Ukovich, 1994), and the IRP with heterogeneous fleet (Chien et al., 1989; Christiansen, 1999; Persson and Göthe-Lundgren, 2005), among others.

When the customers and the vendor belong to the same corporation, it can be beneficial to consider the possibility of transshipments, a situation where customers may ship goods between each other when someone has excess and someone else faces a shortage. This option makes it considerably more difficult to solve the problem, because instead of solving it for the three above-mentioned questions, one must also consider the possibility of transshipping, as a way to save money on inventory (placing less inventory in the system would be possible since shortage risks would be shared) and on routing (smaller quantities would need to be shipped from the supplier).

The purpose of this chapter is to develop a comprehensive review of the related IRP literature, with attention to the opportunities offered by transshipments. It contributes to the current state of knowledge by offering a broader review than some of the recent review papers (Moin and Salhi, 2007; Cordeau et al., 2007; Bertazzi et al., 2008; Andersson et al., 2010). Specifically, we present some formulations and algorithms used to solve the problem. We also suggest that the integration of transshipments into the IRP may lead to savings and add flexibility to the decision maker.

The remainder of the chapter is organized as follows. Sections 2.2 and 2.3 formally define the deterministic IRP and the stochastic IRP, respectively. Section 2.4 is dedicated to some mathematical formulations of the problem, while Section 2.5 points to different solution approaches. Section 2.6 presents the literature review of transshipments in inventory-routing. Some discussions about the relevant literature and our conclusions are made in Section 2.8.

2.2 The Deterministic Inventory-Routing Problem

We now formally introduce the IRP. The problem is defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ where $\mathcal{V} = \{0, ..., n\}$ is the vertex set and $\mathcal{A}$ is the arc set. Vertex 0 represents the supplier and the vertices of $\mathcal{V}' = \mathcal{V} \setminus \{0\}$ represent customers. Both the supplier and customers incur unit inventory holding costs h_i per period ($i \in \mathcal{V}$), and each customer has an inventory holding capacity C_i. The length of the planning horizon is p and, at each time period $t \in \mathcal{T} = \{1, ..., p\}$, the quantity of product made available at the supplier is r^t. We assume the supplier has enough inventory to meet all the demand during the planning horizon and that inventories are not allowed to be negative. The variables I_0^t and I_i^t are defined as the inventory levels at the end of period t, respectively at the supplier and at customer i. At the beginning of the planning horizon the decision maker knows the current inventory level of the supplier and of all customers (I_0^0 and I_i^0 for $i \in \mathcal{V}'$), and has full knowledge of the demand d_i^t of each customer i for each time period t.

There is a set $\mathcal{K} = \{1, ..., K\}$ of vehicles available with capacity Q_k. Each vehicle is able to perform one route per time period to deliver products from the supplier to a subset of customers. A routing cost c_{ij} is associated with arc $(i, j) \in \mathcal{A}$.

The objective of the problem is to minimize the total inventory-distribution cost while meeting the demand for each customer. The replenishment plan is subject to the following constraints:

- the inventory level at each customer can never exceed its maximum capacity.
- inventory levels are not allowed to be negative.
- the supplier's vehicles can perform at most one route per time period, each

starting and ending at the supplier.

- the vehicles' capacity cannot be exceeded.

The solution to the problem should determine which customers to serve in each time period using which of the supplier's vehicles, how much to deliver to each visited customer as well as which routes to use.

Obviously, the IRP defined above is deterministic and static because consumption rates are fixed and known beforehand. In real life the supplier does not always know in advance exactly how much each customer will consume (stochastic demand), nor is this consumption static (dynamic demand).

2.3 The Stochastic Inventory-Routing Problem (SIRP)

The basic idea behind the SIRP is the same as in the IRP, except that the level of realism and the difficulty of solving the problem are increased, given that some data are known only in a probabilistic sense and realizations of such data are revealed gradually to the decision maker. The unkown data can be the demand, the traveling time, the traveling cost, etc. It is easy to observe that many characteristics of the problem are stochastic in real life. These include demand, traveling times, vehicle loading and unloading times, even the availability of the road network. In the stochastic version of the IRP, instead of knowing the consumption rate for each customer, the supplier knows (or estimates) a probability distribution for customer consumption. In this sense, the problem is no longer deterministic and future demands are uncertain. In the classical version of the SIRP, customer demands are mutually independent.

The stochasticity added to the problem creates a probability that shortages will occur. In order to discourage shortages, a penalty is imposed whenever a customer runs out of stock, and this penalty is usually modeled as being proportional to the amount of unsatisfied demand. Unsatisfied demand is typically considered as lost demand, that is, there is no backlogging. Since decisions are made based on partially available information, decisions can lead to expensive course-correcting measures.

The knowledge of the decision maker with respect to the dynamic problem can vary according to the problem at hand. The data can be completely unknown and periodically revealed, but usually the decision maker knows the information in some statistical way, such as a probability distribution estimated from historical data.

The objective remains the same as in the deterministic case, but is written so as to accommodate the stochastic and unknown future parameters: the supplier wants to choose a distribution policy that minimizes its expected discounted value (revenue minus costs) over the planning horizon, which can be finite or infinite.

Whereas most real problems are indeed stochastic, there still exists plenty of research on deterministic models. Exceptions to this rule are traditionally SIRP-related studies involving the oil and gas industry (Bard et al., 1998; Federgruen and Zipkin, 1984; Moin and Salhi, 2007; Trudeau and Dror, 1992) and maritime applications (Bausch et al., 1998; Christiansen et al., 2004; Ronen, 1993, 2002).

2.4 Mathematical models for the IRP

We now present different formulations used to model the problem. Specifically, we show the linear programming formulation of the IRP in Section 2.4.1, the dynamic programming formulation of the SIRP in Section 2.4.2 and the robust programming formulation for the SIRP in Section 2.4.3.

2.4.1 Linear programming formulation of the IRP

Mixed-integer models for the IRP have been around for decades, but due to the complexity of the problem, it is only recently that a very simple version of the problem was solved to optimality by integer linear programming. Using a branch-and-cut algorithm, Archetti et al. (2007) have solved the deterministic multi-retailer case with a single vehicle using the order-up-to level policy as described in Bertazzi et al. (2002). Computational results show the problem was solved optimally within two hours of CPU time instances with up to 50 retailers for when the time horizon is three periods and up to 30 customers with a six-period horizon. Previously, linear programming has been used as a heuristic tool by Campbell et al. (1998). In this section, we present both formulations and algorithms.

The algorithm proposed by Campbell et al. (1998) consists of a two-phase integer programming approach. In the first phase, the period and quantity to be delivered to each customer are computed. Then, in the second phase customers are put together into routes. Obviously the optimality of the second phase is limited by the choices made in the first phase. The model definition and formulation are as follows. Let d_i denote the constant usage rate of customer i, $L_i^t = \max\{0, td_i - I_i^0\}$ denote a lower bound on the total volume to be delivered to customer i by period t, and $U_i^t = td_i + C_i - I_i^0$ be an upper bound on the total volume that can be delivered to customer i by period t. Then, if q_i^t represents the delivery volume to customer i on period t, to prevent stockouts and exceeded inventory capacity one must ensure that

$$L_i^t \leq \sum_{1 \leq s \leq t} q_i^s \leq U_i^t \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{2.1}$$

The total volume that can be delivered on a single period is constrained by a combination of capacity and time windows. Since vehicles are allowed to make more than one trip per period, a way to model the problem based on the resource constraints follows. Let $\mathcal{R}$ be a set of all possible delivery routes r, T_r the duration of route r (as a fraction of a period), and c_r the cost of executing route r. Let x_r^t be a binary variable indicating whether route r is used in period t or not, and q_{ir}^t be a continuous variable representing the delivery volume to customer i on route r in period t. Let Q denote the vehicle capacity and m the time available for a vehicle to perform its routes in a single period.

The problem can then be formulated as follows.

$$\text{minimize} \sum_t \sum_r c_r x_r^t \tag{2.2}$$

subject to

$$L_i^t \leq \sum_{1 \leq s \leq t} \sum_{r: i \in r} q_{ir}^t \leq U_i^t \qquad i \in \mathcal{V} \quad t \in \mathcal{T} \tag{2.3}$$

$$\sum_{i: i \in r} q_{ir}^t \leq Q x_r^t \qquad r \in \mathcal{R} \quad t \in \mathcal{T} \tag{2.4}$$

$$\sum_r T_r x_r^t \leq m \qquad t \in \mathcal{T}. \tag{2.5}$$

Constraints (2.4) ensure that the vehicle capacities are not exceeded, while constraints (2.5) ensure that the time available to perform the routes are sufficient. This model is difficult to solve due to the high number of possible routes, and also because of the length of the planning horizon. Considering a small set of routes and aggregating periods towards the end of the horizon makes the model more computationally efficient. The output of this first phase specifies how much to deliver to each customer in each period of the planning horizon. This information becomes the input of a standard algorithm for the Vehicle Routing Problem with Time Windows which is solved for each period in the second phase. Since decisions are taken separately in the two phases, the second phase can only be optimal with respect to the solution obtained from phase one. In other words, their integration may not be optimal. Besides, this model takes good care of time constraints but does not include any consideration for the inventory holding costs.

The first solvable model for a reasonably sized IRP to optimality was developed by Archetti et al. (2007). It considers a single vehicle to serve all customers, and the replenishment policy is the order-up-to level, as described in Bertazzi et al. (2002). The model works with the following binary variables: x_{ij}^t is equal to 1 if and only if customer j immediately follows customer i on the route of the supplier's vehicle in period t. Let the quantity of product delivered from the supplier to each customer i at each time period t be q_i^t, and let z_0^t be a binary variable equal to one if and only if there is a route to perform in that period. Finally, let z_i^t be a binary variable equal to one if the retailer i is served at time t, and zero otherwise. The problem formulated by Archetti et al. (2007) consists in minimizing the following objective function:

$$\text{minimize} \sum_{t \in \mathcal{T}} h_0 I_0^t + \sum_{i \in \mathcal{V}'} \sum_{t \in \mathcal{T}} h_i I_i^t + \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} \sum_{t \in \mathcal{T}} c_{ij} x_{ij}^t, \tag{2.6}$$

subject to the following constraints:

1. Inventory at the supplier. The inventory level at the supplier at the end of period t is given by its previous inventory level (period $t-1$), plus the quantity r^t made available in period t, minus the total quantity shipped to the customers using the supplier's vehicle in period t:

$$I_0^t = I_0^{t-1} + r^t - \sum_{i \in \mathcal{V}'} q_i^t \qquad t \in \mathcal{T}. \tag{2.7}$$

2. Stockout at the supplier. These constraints impose that the supplier's inventory cannot be negative:

$$I_0^t \geq 0 \qquad t \in \mathcal{T}. \tag{2.8}$$

3. Inventory at the customers. Likewise, the inventory level at each retailer in period t is given by its previous inventory level in period $t-1$, plus the quantity q_i^t delivered by the supplier's vehicle in period t, minus its demand in period t, that is:

$$I_i^t = I_i^{t-1} + q_i^t - d_i^t \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{2.9}$$

4. Stockout at the customers. These constraints guarantee that for each customer $i \in \mathcal{V}'$ the inventory level I_i^t remains non-negative at all time:

$$I_i^t \geq 0 \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{2.10}$$

5. Quantities delivered. These sets of constraints ensure that the quantity delivered by the supplier's vehicle to each customer $i \in \mathcal{V}'$ in each period $t \in \mathcal{T}$ will

fill the customer's inventory capacity if the customer is served, and will be zero otherwise:

$$q_i^t \geq C_i z_i^t - I_i^{t-1} \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{2.11}$$

$$q_i^t \leq C_i - I_i^{t-1} \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{2.12}$$

$$q_i^t \leq C_i z_i^t \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{2.13}$$

If customer i is not visited in period t, then constraints (2.13) mean that the quantity delivered to it will be zero (while constraints (2.11) and (2.12) are still respected). If, otherwise, customer i is visited in period t, then constraints (2.13) limit the quantity delivered to the customer's inventory holding capacity, and this bound is tightened by constraints (2.12), making it impossible to deliver more than what would exceed this capacity. Constraints (2.11) model the OU replenishment policy, ensuring that the quantity delivered will be exactly the bound provided by constraints (2.12).

6. Vehicle capacity: these constraints guarantee that the vehicle's capacity is not exceeded:

$$\sum_{i \in \mathcal{V}'} q_i^t \leq Q \qquad t \in \mathcal{T}. \tag{2.14}$$

7. Routing: these constraints guarantee that a feasible route is determined to visit all customers served in period t:

$$\sum_{i \in \mathcal{V}} q_i^t \leq Q z_0^t \qquad t \in \mathcal{T}; \tag{2.15}$$

$$\sum_{j \in \mathcal{V}', j<i} x_{ij}^t + \sum_{j \in \mathcal{V}', j>i} x_{ji}^t = 2 z_i^t \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{2.16}$$

$$\sum_{i \in \tau} \sum_{j \in \tau, j<i} x_{ij}^t \leq \sum_{i \in \tau} z_i^t - z_k^t \qquad \tau \subseteq \mathcal{V} \quad t \in \mathcal{T}. \tag{2.17}$$

8. Integrality and nonnegativity:

$$q_i^t \geq 0 \qquad i \in \mathcal{V} \quad t \in \mathcal{T}; \tag{2.18}$$

$$x_{ij}^t \in \{0, 1\} \qquad i, j \in \mathcal{V}, i \neq j \quad t \in \mathcal{T}; \tag{2.19}$$

$$x_{i0}^t \in 0, 1, 2 \qquad i \in \mathcal{V} \quad t \in \mathcal{T}; \tag{2.20}$$

$$z_i^t \in 0, 1 \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{2.21}$$

If one's intention is to solve the IRP without the order-up-to level policy, then one should remove constraints (2.11)−(2.13), or alternatively include constraints (2.12) in order to have a maximum allowed inventory level at each retailer. Archetti et al. (2007) have also derived some valid inequalities to make the model more efficient, and were able to solve instances with up to 50 retailers in a three-period horizon.

This model, despite considering a single vehicle to serve the customers, is somewhat more general than others because it considers not only inventory holding costs at the customers, but also at the supplier. This model was later improved by Solyalı and Süral (2011) by using a stronger formulation and a heuristic to provide an upper bound to the branch-and-cut algorithm.

2.4.2 Dynamic programming formulation of the SIRP

A dynamic programming model for the SIRP was introduced by Campbell et al. (1998), where only transportation and stockout costs are taken into account. To simplify the model, no inventory holding costs are incurred. At the beginning of each period the supplier knows the inventory level at each of the customers and decides which customers to visit, how much to deliver to each of them, how to combine them into routes and which routes to assign to each of the available vehicles. The components of their Markov decision process are the following:

- The state x is the current inventory at each customer and the state space $\mathcal{X}$ is $[0, C_1] \times [0, C_2] \times \ldots \times [0, C_n]$. Let $X_t \in \mathcal{X}$ denote the state at time t.

- The action space $\mathcal{A}(x)$ for each state x is the set of all itineraries that satisfy the tour constraints (such as vehicle capacities and customer inventory capacities). Let $\mathcal{A} \equiv \bigcup_{x \in \mathcal{X}} \mathcal{A}(x)$ denote the set of all possible itineraries. Let $A_t \in \mathcal{A}(X_t)$ denote the itinerary chosen at time t.

- The Markov transition function Q obtained from the known demand probability distribution. For any state $x \in \mathcal{X}$ and any itinerary $a \in \mathcal{A}(x)$ the transitions follow
$$P[X_{t+1} \in B \mid X_t = x, A_t = a] = \int\limits_B Q[dy \mid x, a]. \tag{2.22}$$

- The only costs taken into account are transportation costs, which depend on the vehicle tours, and a stockout penalty cost. Let $c(x, a)$ denote the expected daily cost if itinerary $a \in \mathcal{A}(x)$ is chosen and the process is in state x.

- Let $\alpha \in [0, 1)$ denote the discount factor. The objective is to minimize the expected total discounted cost over an infinite horizon. Let $V^*(x)$ denote the optimal expected cost given that the initial state is x, i.e.,

$$V^*(x) \equiv \inf_{\{A_t\}_{t=0}^{\infty}} E\left[\sum_{t=0}^{\infty} \alpha^t c(X_t, A_t) \mid X_0 = x\right]. \tag{2.23}$$

The actions are restricted in the sense that A_t depends only on the history of the system; when one decides which itinerary to choose, one does not know what the future holds. Under certain usual conditions, equation (2.23) can be written as

$$V^*(x) \equiv \inf_{a \in \mathcal{A}(x)} \left\{ c(x, a) + \alpha \int_X V^*(y) Q[dy \mid x, a] \right\}. \tag{2.24}$$

Equation (2.24) can only be solved using classical dynamic programming algorithms if the state space $\mathcal{X}$ is small, which is not the case for practical instances of the SIRP. Campbell et al. (1998) state that it is possible to solve the problem by approximating the value function $V^*(x)$ with a function $\hat{V}(x, \beta)$, where β is a vector of parameters.

This is the approach followed by Kleywegt et al. (2002, 2004) who, as in Campbell et al. (1998) use a Markov decision process to formulate the SIRP. Here, n customers must be served from a warehouse, using m homogeneous vehicles of capacity C_v. Each customer i has an inventory capacity C_i, and the problem is modeled in discrete time $t = 0, 1, 2, \ldots$, usually days. Inventory at each customer i at any given time t is denoted X_i^t and is known to the supplier. Customer demands are stochastic and independent from each other, and the supplier knows the joint probability distribution of their demands, which does not change with time. The supplier must decide which customers to visit, how much to deliver to their local inventories, how to combine customers into routes and which routes to assign to each vehicle. The set of admissible decisions is constrained by vehicle and customer capacities, driver working hours, possible time windows at the customers, and by any other constraint imposed by the system or the application.

Although demands are stochastic, the cost of each decision is known to the supplier. Thus, Kleywegt et al. (2002, 2004) define the following costs:

- Traveling costs c_{ij} on the arcs (i, j) of the network.
- Shortages, if they occur, are proportional to the amount of unsatisfied demand s_i at customer i and cost $s_i(p_i)$. In this model unsatisfied demand is lost.
- Inventory holding costs are incurred on the existing inventory x_i at customer i plus the amount delivered to this customer, q_i, and are equivalent to $(x_i + q_i)h_i$.

- Finally, if the supplier delivers q_i at customer i, then he earns a revenue of $r_i(q_i)$.

The problem is formulated so as to maximize the expected discounted value over an infinite horizon as a discrete time Markov decision process as follows. Let X_{it} denote the inventory level at customer i at time t. Thus x is the current inventory at each customer and the state space $\mathcal{X}$ is $[0, C_1] \times [0, C_2] \times \ldots \times [0, C_n]$. Let $X_t = (X_{1t}, X_{2t}, \ldots, X_{nt})$ denote the state at time t. The action space $\mathcal{A}(x)$ for each state x is the set of feasible decisions, that is, the ones that satisfy the constraints of the problem such as vehicle and customer capacities and any other constraint needed. Let $A_t \in \mathcal{A}(X_t)$ denote the decision made at time t. Let $k_{ij}(a)$ denote the number of times that arc (i, j) is traversed while executing decision a, for any a and arc (i, j). Finally, for any customer i, let $q_i(a)$ denote the quantity delivered to customer i while executing decision a.

Let d_{it} denote the demand at customer i at time t. Since there is no backlogging, the usage cannot be higher than the amount available. In the way Kleywegt et al. (2002) formulate the problem, the customer's inventory plus the amount delivered are available for use in the same period. Thus the amount of product used by customer i at any time t is given by $\min\{d_{it}, X_{it} + q_i(A_t)\}$ and the shortage at customer i at any time t is $S_{it} = \max\{0, d_{it} - (X_{it} + q_i(A_t))\}$.

Recently, Bertazzi et al. (2012) have also proposed a dynamic programming model for the SIRP, even though the model is later solved heuristically. This is due to the fact that the problem is more general than that studied in Kleywegt et al. (2002) and Kleywegt et al. (2004), especially with respect to the routing aspect. Kleywegt et al. (2002) study the case with direct deliveries only, Kleywegt et al. (2004) limits the routing to at most three customers per route, whereas Bertazzi et al. (2012) consider the more general case.

2.4.3 Robust programming formulation of the SIRP

Robust optimization is an approach to deal with uncertainty where, in the extreme case, no information is available of the parameter probability distributions. This is done by optimizing the problem ensuring feasibility for all possible realizations of the bounded uncertain parameters, also called a minimax solution. Usually studies on the SIRP assume one knows the probability distribution of demand, which is generally not true. Such probability distribution has to be somehow estimated or forecast in order to use the models just described. Robust optimization can be used to cope with real-life situations where one does not have that information in advance.

Solyalı et al. (2012) propose such an approach, which will be detailed in this section.

Their model presents the following description: a supplier distributes a single product to n customers, using a vehicle of capacity C, over a finite discrete time horizon p. The dynamic uncertain demand at each customer $i \in \mathcal{V} = \{1, \dots, n\}$ in period $t \in \mathcal{T} = \{1, \dots, p\}$ is d_{it}. The probability distribution of the demand is unknown, but one knows that it can take any value in the interval $[\bar{d_{it}} - \hat{d_{it}}, \bar{d_{it}} + \hat{d_{it}}]$, where $\bar{d_{it}}$ is the nominal value (point estimate), and $\hat{d_{it}}$ is the maximum deviation for the demand of i in period t. An inventory holding cost is incurred at the customers, equal to h_{it} per unit at customer i in period t. Backlogging is allowed and each unit backlogged in period t at customer i costs g_{it}, where $g_{it} > h_{it}$. There is a fixed vehicle dispatching cost f_t for using the vehicle in period t. If the vehicle leaves customer $i \in \mathcal{V}' = \mathcal{V} \cup \{0\}$ heading to customer j it incurs a cost c_{ij}, and these transportation costs are symmetric.

The problem is formulated as follows. Let q_{itk} be the total inventory cost of replenishing customer i in period $t \in \mathcal{T}$ to satisfy its demand in period $k \in \mathcal{T}$; $q_{i,T+1,k}$ be the total inventory cost of not meeting the demand of customer i in period $k \in \mathcal{T}$; let w_{itk} be the fraction of the demand of customer i in period $k \in \mathcal{T}$ delivered in period $t \in \mathcal{T}$; and let $w_{i,T+1,k}$ be the fraction of the unsatisfied demand of customer i in period $k \in \mathcal{T}$. Additionally let y_{it} be 1 if customer i is replenished in period $t \in \mathcal{T}$ and 0 otherwise; y_{0t} be 1 if the vehicle is used in period $t \in \mathcal{T}$ and 0 otherwise; and $x_{ijt}(i > j)$ be the number of times the edge (i, j) is traversed in period $t \in \mathcal{T}$. Then, the robust IRP can be formulated as follows:

$$\text{minimize} \sum_{t \in \mathcal{T}} f_t y_{0t} + \sum_{i \in \mathcal{V}'} \sum_{j \in \mathcal{V}', j<i} \sum_{t \in \mathcal{T}} c_{ij} x_{ijt} + \sum_{i \in \mathcal{V}} \sum_{t=1}^{p+1} \sum_{k=1}^{p} d_{ik} q_{ikt} w_{itk} \tag{2.25}$$

subject to

$$\sum_{t=1}^{p+1} w_{itk} = 1 \qquad i \in \mathcal{V} \quad k \in \mathcal{T}; \tag{2.26}$$

$$w_{itk} \leq y_{it} \qquad i \in \mathcal{V} \quad t, k \in \mathcal{T} \quad d_{ik} > 0; \tag{2.27}$$

$$\sum_{i \in \mathcal{V}} \sum_{k=1}^{p} d_{ik} w_{itk} \leq C y_{0t} \qquad t \in \mathcal{T}; \tag{2.28}$$

$$\sum_{j \in \mathcal{V}', j<i} x_{ijt} + \sum_{j \in \mathcal{V}', j>i} x_{jit} = 2 y_{it} \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{2.29}$$

$$\sum_{i \in S} \sum_{j \in S, j<i} x_{ijt} \leq \sum_{i \in S} y_{it} - y_{kt} \qquad S \subseteq \mathcal{V} \quad t \in \mathcal{T} \quad k \in S; \tag{2.30}$$

$$y_{it} \leq y_{0t} \qquad i \in \mathcal{V} \quad t \in \mathcal{T}; \tag{2.31}$$

$$x_{ijt} \in \{0,1\} \qquad i,j \in \mathcal{V}, j < i \quad t \in \mathcal{T}; \tag{2.32}$$

$$x_{i0t} \in \{0,1,2\} \qquad i \in \mathcal{V} \quad t \in \mathcal{T}; \tag{2.33}$$

$$y_{it} \in \{0,1\} \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{2.34}$$

$$w_{itk} \geq 0 \qquad i \in \mathcal{V} \quad k \in \mathcal{T} \quad 1 \leq t \leq p+1, \tag{2.35}$$

where $q_{itk} = \sum_{l=t}^{k-1} h_{il}$ if $t \leq k$ and $q_{itk} = \sum_{l=k}^{t-1} g_{il}$ if $t > k$.

The objective function (2.25) is the sum of the fixed vehicle dispatching, transportation, inventory holding and shortage costs. Constraints (2.26) specify that the demand of customer i in period k is either met from periods 1 through p, or lost. Constraints (2.27) allow the vehicle to serve customer i in period t only if a replenishment to customer i takes place in period t. Contraints (2.28) ensures that the vehicle capacity is not exceeded. Constraints (2.29) are degree constraints, guaranteeing that if i is visited in period t, then there are two edges incident to it. Constraints (2.30) are subtour elimination constraints. Constraints (2.31) ensure the vehicle starts its tour from the supplier. Constraints (2.32)−(2.34) and (2.35) are integrality constraints and nonnegativity constraints, respectively.

If d_{ik} is replaced by $\bar{d_{it}}$ for $i \in \mathcal{V}$, $t \in \mathcal{T}$, then it is called the nominal formulation, since it does not incorporate any robustness. It is not trivial to derive the robust formulation, and the reader is referred to Solyalı et al. (2012) for details. Their final robust formulation ensuring feasibility for any $d_{ik} \in [\bar{d_{it}} - \hat{d_{it}}, \bar{d_{it}} + \hat{d_{it}}]$ is

$$\text{minimize} \sum_{t \in \mathcal{T}} f_t y_{0t} + \sum_{i \in \mathcal{V}'} \sum_{j \in \mathcal{V}', j<i} \sum_{t \in \mathcal{T}} c_{ij} x_{ijt} + \sum_{i \in \mathcal{V}} \sum_{t=1}^{p+1} \sum_{k=1}^{p} q_{itk} w'_{itk} \tag{2.36}$$

subject to (2.29)−(2.34) and to

$$\sum_{i \in \mathcal{V}} \sum_{k=1}^{p} w'_{itk} \leq C y_{0t} \qquad t \in \mathcal{T}; \tag{2.37}$$

$$w'_{itk} \geq 0 \qquad i \in \mathcal{V} \quad k \in \mathcal{T} \quad 1 \leq t \leq p+1; \tag{2.38}$$

$$\sum_{t=1}^{p+1} w'_{itk} \geq \bar{d_{it}} + \hat{d_{it}} \qquad i \in \mathcal{V} \quad k \in \mathcal{T}; \tag{2.39}$$

$$w'_{itk} \leq (\bar{d_{it}} + \hat{d_i t}) y_{it} \qquad i \in \mathcal{V} \quad t \in \mathcal{T} \quad k \in \mathcal{T}, \tag{2.40}$$

where $w'_{itk} = d_{ik} w_{itk}$. Using this formulation the authors have solved instances with up to seven periods and 30 customers within a reasonable computing time.

2.5 Algorithms for the IRP

There exist so many algorithms for the IRP that it is difficult to find two authors solving the problem in exactly the same way. Due to this high fragmentation, even categorizing the literature becomes a hard task. The assumptions and definitions are almost unique for each one of the papers reviewed. In addition, most real-life IRPs are either dynamic, stochastic, or work with a long term horizon. These three characteristics add enormous difficulty to the problem, and common assumptions involve static demand (be it deterministic or stochastic) and a short term solution.

Previous literature reviews have analyzed the IRP from different standpoints. For instance, Baita et al. (1998) study what they call Dynamic Routing-and-Inventory Problems, which are "characterized by the simultaneous relevance of routing and inventory issues in a dynamic environment". Since they consider the dynamics of the problem, they are concerned with time: whether decisions are taken on the frequency domain, that is, decision variables are replenishment frequencies, or headways between shipments, or conversely decisions are taken on the time domain − schedule of shipments, routes and quantities.

Campbell et al. (1998) prefer to view the IRP through the solution approaches developed to solve it. Moin and Salhi (2007) sort the papers according to their planning horizon (single period, multi period and infinite horizon), but classify the SIRP separately. Andersson et al. (2010) also classify according to the planning horizon, but include both stochastic and deterministic cases inside this arrangement.

Due to the difficulty of this problem, which is obviously NP-hard as it contains the CVRP, most papers propose heuristics and especially metaheuristics to solve it. To cite a few, Ribeiro and Lourenço (2003) proposed an Iterated Local Search heuristic, Zhao et al. (2008) developed a Variable Neighborhood Search mechanism, Campbell and Savelsbergh (2004) put forward a Greedy Randomized Adaptive Search scheme and Boudia and Prins (2009) presented a memetic algorithm.

We organize our literature review on algorithms in three parts. In the first we discuss the classical configuration with deterministic demand and one product. The second part will present algorithms for one product with stochastic demand, while the last part will present an overview of several other configurations, including multi-product, split deliveries, among others. After each section we present a table summarizing the main papers in the area, organized by authors and date. At the end a different table is presented, summarizing all papers mentioned in previous tables, but organized by their features.

2.5.1 Deterministic case, one product

In the early 1980s some studies have started to incorporate inventory concerns within the existing vehicle routing literature. These were mostly variations of VRP models and heuristics developed to accommodate inventory costs. The first such paper is due to Bell et al. (1983), which was followed by Federgruen and Zipkin (1984), Blumenfeld et al. (1985), Burns et al. (1985), Dror et al. (1985), Dror and Levy (1986), Dror and Ball (1987), Anily and Federgruen (1990). Most of those papers considered consumption rate at the customers as known and deterministic. Despite the large number of papers on distribution and inventory matters before this period, the combination of these two features remained very difficult to solve, not only because of computer power, but also because of the algorithms developed for large and complex combinatorial problems, such as the ones involving both distribution systems and inventory management optimization at the same time.

For instance, Bell et al. (1983) have analyzed the case where only transportation costs are included, but inventory levels must be met at the customers. Since in their case original demands were stochastic (they used forecasts to make them seem deterministic to the model), more details will be provided in the next section.

A short term solution is presented in Dror et al. (1985) and in Dror and Ball (1987) (who also studied the stochastic version), based on the assignment of customers to so-called optimal replenishment periods, and then calculating the expected increase in cost if the customer is visited in another period. Dror et al. (1985) offered the first algorithmic comparison for the IRP. In their paper, they handle the problem with two major simplifications: (1) once a customer is visited, the amount of product delivered fills the customer's capacity (order-up-to level policy), and (2) customers are only visited once during the planning period (e.g. one week). They create two subsets out of the customers set, one containing customers that must be visited, the other containing customers that could be visited. They solve the problem in two phases as follows. For the customers that must be visited, they calculate the costs of visiting the customer earlier than the latest period possible. For customers that could be visited, they compute the future cost difference between visiting or not this customer. Based on these costs, customers are assigned to periods, and VRPs are solved for each period, followed by a node interchange improvement. They considered a deterministic version of the problem, and proposed two algorithmic solutions. The first one assigns customers to periods in a first step, and then solves a VRP for each period. The other first assigns customers not only to periods, but also to vehicles, so that the second part needs only solve one TSP for each period and each vehicle. In both cases an integer program is solved to assign customers to vehicles, minimizing transportation and inventory costs. The second part of algorithm works with the output of the first part, which was obtained heuristically and there is no guarantee of its quality.

Building on the idea of adapting previous VRP algorithms and heuristics, Dror and Levy (1986) have proposed a node interchange algorithm for a weekly IRP. They have generated an initial solution to a VRP, keeping track of vehicle capacities and customer inventories, improving the initial solution presented in Dror et al. (1985).

Burns et al. (1985) have developed formulas based on the trade-offs between transportation and inventory costs, using an approximation of traveling costs. They show that when using direct shipping the optimal shipment size is the Economic Order Quantity (EOQ). In order to serve many customers within one route, the vehicle must carry a full truckload, with the trade-off being influenced by the number of customers served in the route, due to the use of approximations of the local distance traveled within delivery regions.

Dror and Ball (1987) simplify the problem by fixing the amount delivered to each customer in order to fill up its inventory capacity (order up-to-level policy), and in this sense, the amount delivered only depends on the period of the delivery since their approach is deterministic. They have simplified real stochastic demands to a deterministic approach using three different variables: one to penalize early deliveries to customers with sufficient inventory, another to motivate deliveries to customers that are not restrictive (not required by the constraints), and the last one to identify customers that must be served within the planning horizon. Finally, their solution involves assigning customers to one period of the planning horizon through a generalized assignment algorithm, solving the VRP for each period of the planning horizon and finally trying to improve the solution by promoting interchanges not only within routes but also within periods.

A different approach is used in Anily and Federgruen (1990) for the deterministic IRP with an infinite horizon. Customers are divided into regions, and whenever a customer is to be visited, all customers within that region are visited as well. Gallego and Simchi-Levi (1990) evaluate the long-term effectiveness of direct shipping on a system with one warehouse and multiple retailers. The direct shipping proved to be 94% effective whenever the vehicle capacity is at least 71% used.

Allowing vehicles to perform more than one route per period, Aghezzaf et al. (2006) have modified the approach used by Anily and Federgruen (1990). Using column generation, new multi-tours (columns) are created using an extension of the savings algorithm. Inventories at the customers are replenished following an EOQ policy. In Raa and Aghezzaf (2009) this work is extended with the addition of some driving time constraints.

Abdelmaguid and Dessouky (2006) have proposed a genetic algorithm to a dynamic deterministic version, which is shown to outperform the previously proposed construction heuristic by Abdelmaguid (2004).

With limited inventory at the warehouse as in Federgruen and Zipkin (1984), Chien et al. (1989) formulate a mixed-integer linear model, considering customer selection, resource allocation and vehicle routing with heterogeneous fleet. A heuristic is used to solve the problem. It first generates vehicles routes based on previously solved inventory allocation and makes customers assignments. An improvement algorithm is then applied. Also allowing backlogging, Abdelmaguid et al. (2009) derive a constructive and improvement heuristic for the multi-period single-item version of the problem. These are two of the few papers to use backlogging for unsatisfied demand.

Some of the above mentioned papers (for instance Anily and Federgruen (1990), Blumenfeld et al. (1985), Burns et al. (1985)) consider continuous decision variables for the delivery times. Under this assumption, the optimal replenishment time may be non-integer, an impractical solution for most suppliers. Roundy (1985) studies the case with several retailers, direct deliveries and discrete time, defining frequency based policies proven to be within 2% of the optimum in the worst case. In this model, costs are linear for inventory holding and fixed for ordering, which includes delivery costs.

Bertazzi et al. (2002) study the case with multiple retailers, still with deterministic demand rates and the order-up-to level policy, decreasing the flexibility of the decision maker, but simplifying the set of possible decisions of the problem. The problem is solved heuristically in steps. The first step creates a feasible solution, and the second one improves it as long as a given minimum improvement is made to the total cost function. This is achieved by removing all possible customer pairs and computing a series of shortage paths to determine the periods in which the customers should be reinserted. They consider both inventory and transportation costs and it is relevant to note that the supplier also incurs inventory costs in their model, which was generally not considered in other papers.

Campbell and Savelsbergh (2004) use the same two-phases idea, but in their allocation problem they group customers into clusters which can be served by one vehicle. The allocation phase then determines how much to deliver to each customer, and a VRPTW is solved for the first periods of the planning horizon.

Savelsbergh and Song (2008) solve a problem in which a single producer cannot usually meet the demand of its customers, and customers who are so far away that the tours cannot fit in one period. This leads to the formulation of a problem with several suppliers, who must be visited one after another, in trips lasting longer than one period. This problem is called the IRP with continuous moves and is solved through a local search algorithm applied on a solution obtained by a randomized greedy heuristic.

Within a cyclic planning context in which demands are deterministic and therefore a long-term distribution pattern can be derived, Raa and Aghezzaf (2008) develop an algorithm allowing vehicles to perform multiple tours, possibly with different frequencies. Customers are partitioned over vehicles using a column generation algorithm, and the actual assignment of customers to tours is done by a greedy heuristic. Tests are performed on random instances containing up to 100 customers.

Maritime applications of the ship routing and inventory management can be found in Christiansen (1999), and in Christiansen and Nygreen (1998a,b), among others. Deterministic cases are studied and solved using Lagrangian relaxation. These problems show the special feature of having several suppliers as well as several retailers (many-to-many structure), exploring continuous routing.

The first paper to offer an exact algorithm to the IRP is due to Archetti et al. (2007). These authors considered the case with only one vehicle, no backlogging and using the order up-to-level inventory policy. They developed a branch-and-cut and derived several valid inequalities for the problem. Also using a branch-and-cut algorithm, Solyalı and Süral (2011) have improved their results with an exact mixed-integer programming formulation and developed a MIP based heuristic for the problem.

When the planning horizon is not limited and one is concerned with solving the problem over the long-run (infinite horizon problem), the objective can no longer be the minimization of total costs over the horizon, and in such a case two approaches have been used (Andersson et al., 2010). The first one considers the minimization of average daily costs using repeatable policies. For instance, this is the approach used in Anily and Federgruen (1990). The alternative is a full dynamic programming formulation, which uses a discount factor, decreasing the importance of costs and revenues associated with later time periods. Kleywegt et al. (2002, 2004); Adelman (2004), among others, use this approach, which will be seen in details in the next section.

With the aim of finding the Pareto-optimal solutions, Geiger and Sevaux (2011b) plan the tactical level of the IRP, compare different solutions with respect to the two opposing terms in the objective function. When a customer is visited very often, its inventory cost is low but routing becomes expensive, and vice versa. This is important when considering changes in one of the terms, for example when fuel prices change or when focusing towards "green" logistics solutions.

A comparison of two MILP formulations is proposed by Aksen et al. (2012) for an application of the IRP related to waste vegetable oil collection. These authors derive lower bounds by partially relaxing integrality in their model, and observe that the solutions obtained are on average within 3.28% of optimality.

Finally, a powerful hybrid heuristic is presented in Archetti et al. (2012). It operates with a combination of a tabu search embedded within four neighborhood searches and two MIPs to further refine the solutions. Their results show that the heuristic performs very well on benchmark instances, with an optimality gap usually below 0.1%.

Table 2.2 presents the main papers cited in this section.

2.5.2 Stochastic case, one product

The IRP becomes more realistic when one considers that customers have a stochastic demand instead of a fixed usage rate. As a result, linear programming is no longer the preferred approach to solve this version of the problem, dynamic programming being the choice of most researchers.

Bell et al. (1983) have proposed a linear programming model to solve a deterministic simplification of the problem. They use heuristics to generate forecasts of the unknown demand. Possible delivery routes are created heuristically, and continuous variables represent the amount to be delivered. Their mixed-integer programming formulation used about 100,000 to 200,000 binary variables, 300,000 to 600,000 continuous variables and about 100,000 to 200,000 constraints. Using Lagrangian relaxation, they were able to decompose the problem into one knapsack problem for each vehicle and their solution was proved to be within 2% of the optimum. However, the heuristic used to generate forecasts was very simple, based on a simple exponential smoothing model tested with only 10 different values for the smoothing parameter.

Federgruen and Zipkin (1984) have modified the VRP heuristic proposed by Fisher and Jaikumar (1981) to accommodate inventory and shortage costs in a random demand environment. Stochasticity is treated within the objective function which is nonlinear and contains three terms: the routing cost, the assignment of customers to vehicles, and the amount delivered to each of them. Their heuristic, based on fixing the assignment of customers to one of the vehicles from the heterogeneous fleet, generated good solutions within reasonable computing time, since the

Table 2.2: Classification of main papers on single item deterministic IRP

Authors	**Year**	**Time horizon**		**Structure**			**Routing**			**Inventory policy**			**Fleet composition**		**Fleet size**		
		Finite	Infinite	One-to-one	One-to-many	Many-to-many	Direct	Multiple	Continuous	Lost sales	Backlogging	Non-negative	Homogeneous	Heterogeneous	Single	Multiple	Unconstrained
Dror et al.	1985	✓			✓			✓		✓			✓			✓	
Dror and Ball	1987	✓		✓	✓			✓		✓			✓			✓	
Chien et al.	1989	✓			✓			✓			✓			✓		✓	
Anily and Federgruen	1990		✓		✓			✓		✓			✓				✓
Gallego and Simchi-Levi	1990		✓		✓		✓			✓			✓				✓
Christiansen	1999	✓				✓			✓	✓				✓		✓	
Bertazzi et al.	2002	✓			✓			✓				✓	✓		✓		
Adelman	2003		✓		✓			✓		✓			✓			✓	
Campbell and Savelsbergh	2004	✓			✓			✓		✓			✓			✓	
Aghezzaf et al.	2006		✓		✓			✓		✓			✓			✓	
Archetti et al.	2007	✓			✓			✓				✓	✓		✓		
Savelsbergh and Song	2008	✓				✓			✓	✓			✓			✓	
Abdelmaguid et al.	2009	✓			✓			✓			✓			✓		✓	
Raa and Aghezzaf	2009		✓		✓			✓		✓			✓			✓	
Archetti et al.	2012	✓			✓			✓				✓	✓		✓		
Aksen et al.	2012	✓			✓			✓				✓	✓		✓		

problem is decomposed into an inventory allocation problem and a TSP for each vehicle. They also derived an exact algorithm for the problem using generalized Benders decomposition.

Using a different approach, Golden et al. (1984) determine which customers to visit before solving the routing problem. Based on degrees of urgency, all customers with inventory below a given threshold are considered as potential customers to be visited. A modified TSP with time constraints is then solved in order to decide which customers to actually visit, depending on their urgency. The selected ones are put into routes by solving a VRP with the Clarke and Wright (1964) algorithm, and if the VRP is not feasible due to time constraints, the preceding TSP is solved again with tighter constraints.

As opposed to what Dror et al. (1985) did for the deterministic IRP, i.e. eventually including a customer in a route even if it was not its optimal replenishment period, Bard et al. (1998) determine the optimal replenishment interval for each customer. If the optimal replenishment time of a customer falls outside the planning horizon being considered, then the latter is extended to include this customer. Their model considers random consumption and there are available satellite facilities where the vehicles can refill. Modifying an earlier paper by Dror et al. (1985), Trudeau and Dror (1992) introduce unknown stochastic demands and route failures when the load of the vehicle is insufficient to serve the next customers in the route. The problem is solved heuristically by selecting possible customers to visit, assigning them to periods and routing vehicles using the Clarke and Wright heuristic. Inter- and intra-period improvements are performed afterwards. The authors compare three different approaches for the selection of the customers' replenishment periods.

Aghezzaf (2008) studies the case with normally distributed customer demands and travel times with constant averages and bounded standard deviations. He uses robust optimization to determine the distribution plan through a non-linear mixed-integer programming formulation which is feasible for all possible realizations of demand and travel times. Monte-Carlo simulation is used to improve the plan's critical parameters (replenishment cycle times and safety stock levels).

Given the size and the complexity of the SIRP, Minkoff (1993) proposes a heuristic approach based on a Markov decision model to a problem somewhat similar to the IRP, called the Dispatch Delivery Problem. He simplifies the value function, making it a sum of smaller value functions, one for each customer, and solves the problem heuristically. This model is one of the few to work with an unconstrained fleet. Berman and Larson (2001) also use dynamic programming to solve the case where the demand probability distributions are known, adjusting the amount of goods delivered to each customer, in order to minimize the expected sum of delivering too early or too late, of delivering less than the customer's capacity and of traveling back to the depot with the remaining products.

Also using dynamic programming, Kleywegt et al. (2002) include constraints for the number of vehicles available, and allow only direct deliveries. Immediate reward functions are composed of individual customer immediate rewards (revenue minus the sum of travel, inventory and shortage costs). The optimal expected value is the total discounted sum of all rewards. Since there are more customers than vehicles, one should decide which customers to serve. Their dynamic programming algorithm is shown to outperform linear programming on all instances. This work was later extended by Kleywegt et al. (2004) to handle multiple deliveries per trip, allowing up to three deliveries per route. In Adelman (2004) there is no limit to the number of customers to be served in a route, except for the limits resulting from maximal route duration and vehicle capacity. His approach is a little different and works as follows. Using a value function not made up of individual customer values, but of marginal transportation costs, Adelman (2003, 2004) compare stockout costs with replenishment policies, choosing the one that maximizes the value. A linear program is derived from the value function, and its optimal dual prices are used to calculate the optimal policy of the semi-Markov decision process. The deterministic case is solved in Adelman (2003), and the stochastic case is dealt with in Adelman (2004).

Hvattum and Løkketangen (2009) and Hvattum et al. (2009) solve the problem heuristically, capturing the stochastic information over a short horizon. In Hvattum and Løkketangen (2009) the problem is solved using a GRASP which successively increases the volume delivered to customers. Hvattum et al. (2009) state that it is sufficient to capture the stochastics of the SIRP over a finite horizon and they do it through scenario trees. Their heuristic is based on a top-down GRASP which starts at the root node and continues recursively, taking advantage of the knowledge gathered in previous constructions through the principle of marginal conditional validity (Glover, 2000; Hvattum et al., 2005).

Bertazzi et al. (2012) formulate the SIRP using dynamic programming. They solve it approximately using a hybrid rollout algorithm approximating the cost-to-go function with a linear program model.

Geiger and Sevaux (2011a) have studied a problem with unknown demand varying ± 10% around a mean value. They proposed several policies based on delivery frequencies for each customer. They provide the Pareto front approximation of such policies when moving from a total routing-optimized solution to an inventory-optimized one. In order to solve the problem for many periods, they apply the

Record-to-Record Travel heuristic of Li et al. (2007).

Finally, column generation is applied to the tactical solution of an IRP in Michel and Vanderbeck (2012). In this study, customer demands are stochastic and are clustered to be served by different vehciles. Routing costs are approximated, since the actual routing problem is considered to be operational and solved at a later stage. The proposed method yields solutions within approximately 6% of optimality.

Table 2.3 presents the main papers cited in this section.

2.5.3 Other variants

As stated earlier, there exist numerous variations of the IRP. In this section we will mention the most important ones and some related literature.

Since the VMI provides advantages for both the supplier and retailers, it is natural to think that integrating one more element of the supply chain may lead to an even better performance. This extra element may be external (the supplier of the supplier) or another department of the actual supplier, such as the production level. This leads to the production-inventory-routing problem, also called production-distribution problem. Blumenfeld et al. (1985) consider distribution, inventory and production set-up costs, using three different configurations between suppliers and buyers: (1) direct links, (2) via a consolidation terminal, and (3) a combination of both. Demands and costs are still deterministic and constant in their work. The interested reader is directed to Chandra and Fisher (1994); Fumero and Vercellis (1999); Bertazzi et al. (2005); Bard and Nananukul (2009, 2010) and Archetti et al. (2011). In the same vein, Javid and Azad (2010) have proposed a broader mechanism that simultaneously optimizes location, allocation, capacity, inventory and routing decisions in a supply chain network design under stochastic demands. They propose a mathematical formulation which obviously can only be solved for relatively very small instances. They also propose a heuristics based on a tabu search combined with simulated annealing.

Another variation of the IRP is the one that handles several products at once. Speranza and Ukovich (1994, 1996) study the case with pre-determined frequencies for a multi-product flow on a single link. Bertazzi et al. (1997) later expanded these studies to handle multiple customers. Every customer is analyzed individually, and those with the same optimal frequency are aggregated for the calculation of routes. Federgruen et al. (1986) have extended the work by Federgruen and Zipkin (1984) to allow multiple products, in their case, perishable items. This is done with a penalty for unsold out-of-date items. Carter et al. (1996) have proposed a two-phase heuristic

Table 2.3: Classification of main papers on single item stochastic IRP

Authors	**Year**	**Time horizon**		**Structure**			**Routing**			**Inventory policy**			**Fleet composition**		**Fleet size**		
		Finite	Infinite	One-to-one	One-to-many	Many-to-many	Direct	Multiple	Continuous	Lost sales	Backlogging	Non-negative	Homogeneous	Heterogeneous	Single	Multiple	Unconstrained
Bell et al.	1983	✓			✓			✓		✓				✓		✓	
Federgruen and Zipkin	1984	✓			✓			✓		✓				✓		✓	
Golden et al.	1984	✓			✓			✓		✓			✓			✓	
Dror and Ball	1987	✓		✓	✓			✓		✓			✓			✓	
Trudeau and Dror	1992	✓		✓				✓		✓			✓			✓	
Minkoff	1993		✓		✓			✓		✓			✓				✓
Bard et al.	1998	✓				✓		✓		✓			✓			✓	
Berman and Larson	2001		✓		✓			✓		✓			✓		✓		
Kleywegt et al.	2002		✓		✓		✓			✓			✓				✓
Adelman	2004		✓		✓			✓		✓			✓			✓	
Kleywegt et al.	2004		✓		✓			✓		✓			✓			✓	
Hvattum and Løkketangen	2009		✓		✓			✓		✓			✓			✓	
Hvattum et al.	2009		✓		✓			✓		✓			✓			✓	
Bertazzi et al.	2012	✓			✓			✓		✓			✓		✓		

to solve the multi-product version of the problem. They solve an allocation problem first, where they choose when and how much to deliver to customers, and they then construct the delivery routes. In their work the allocation problem is first solved with capacity constraints equivalent to the total vehicle capacity, followed by a VRPTW to determine the routes. A multi-item IRP with demand uncertainty and a fleet of homogeneous vehicles is studied by Huang and Lin (2010) who solve it by means of an ant colony optimization algorithm. A particular case of the multi-item IRP is analyzed by Popović et al. (2012) in which different types of fuel are delivered to a set of customers by vehicles with compartments. The problem is solved by means of variable neighborhood search heuristic since the proposed MILP can only handle the smallest instance from a practical application.

A variation of the multi-product version, which also considers multiple suppliers but only one customer, was analyzed by Moin et al. (2011). The authors obtain lower and upper bounds after solving a linear mathematical formulation with a commercial solver and then generate better upper bounds by means of a genetic algorithm. The problem is defined with several suppliers, each offering a different product, and with an assembly plant where the items should be delivered by a fleet of homogeneous vehicles to satisfy a known demand. An extension of the previous structure, the many-to-many one was studied by Ramkumar et al. (2012) who proposed a MILP to a multi-item multi-depot IRP. However, computational results show the limitations of the method as instances with only two vehicles, two products, two suppliers, three customers and three periods could not be solved to optimality in eight hours of computing time.

Maritime applications are also common in inventory-routing. A deterministic maritime multi-product IRP, with several loading ports with variable production is considered in Persson and Göthe-Lundgren (2005). The solution method used is based on column generation with the use of valid inequalities. Not limited by frequency-based policies, Bertazzi and Speranza (2002) develop a model for the shipment of many products through a single link, considering both routing and inventory costs, yet with deterministic and constant demand rates. Qu et al. (1999) formulate the stochastic multi-product case with an iterative algorithm, alternating between an inventory problem which determines the points to visit, and a routing problem, to compute routing costs which are fed back to the inventory problem. A similar problem with deterministic demands and vehicle and capacity constraints was solved by Stacey et al. (2007). A tactical approach is studied by Grønhaug et al. (2010). These authors are concerned with the bulk transportation of liquefied natural gas from many liquefaction plants to several regasification terminals using a heterogeneous fleet of specialized ships. In this version of the problem, the load of the ships evaporates at a constant rate, and the supply as well as the demand are variable. A multiple product version of a similar problem is analyzed by Christiansen et al. (2011), who also deal with a many-to-many structure and a heterogeneous fleet of capacitated ships. The largest instance they solve contains five ships, 12 suppliers, 49 customers and 11 products, and is solved over 14 periods. The algorithm proposed by these researchers is based on genetic search. A practical industry problem is studied in Song and Furman (2012) in which an optimization-based heuristic is developed. The problem at hand is deterministic and has a many-to-many configuration. The algorithm developed can solve instances with up to five suppliers, five customers and nine ships over 60 periods. Complicating constraints such as time windows and the cost structure of the problem, which is typical of the maritime environment (i.e. demurage and overage rates), make the algorithm very flexible to handle real-life instances. Stålhane et al. (2012) seek a long term efficient solution in which multiple products are delivered in a one-to-many structure. Given the nature of the problem, described as a liquefied natural gas application, direct deliveries are usually considered. The aim is to find good annual delivery plans for the fleet of heterogeneous ships. A problem in which customers present varying storage capacities as well as multiple production sites where both production and consumption rates are variable is studied by Engineer et al. (2012). The authors propose a branch-and-price-and-cut algorithm. Due to the complexity of the problem, only small instances can be solved. An extension of the fix-and-relax heuristic is applied by Uggen et al. (2011). It works by iteratively solving the problem with integer variables for the first and more important periods, and LP relaxed integer variables for the remaining periods; once this iteration is complete, the first set of variables is then fixed, a new set of variables are set to be integer again and the algorithm iterates. The authors have tested their implementation on LNG instances arising in the maritime IRP context.

The classical road-based IRP is considered in Liu and Lee (2011), who also add time windows. Their algorithm is designed with a combination of variable neighborhood search and tabu search. However, the effectiveness of the algorithm cannot be completely assessed because their comparison is made against three algorithms designed for the VRPTW.

Another specific case is the IRP with direct deliveries, as the one studied by Kleywegt et al. (2002) and by Bertazzi (2008). Direct deliveries simplify the problem since one is not concerned about making efficient routes for several customers. It is shown to be effective when economic order quantities for the customers are close to the vehicle capacities (Gallego and Simchi-Levi, 1990, 1994). Li et al. (2010) have developed an analytic method for performance evaluation of this delivery strategy, whose effectiveness can be represented as a function of system parameters.

Following the work of Savelsbergh and Song (2008) who developed their model to the IRP with continuous moves with several loading and unloading points, Christiansen (1999) and Christiansen and Nygreen (1998a,b) have considered an application of the maritime ship routing and inventory management problem. The problem was formulated and solved using Dantzig-Wolfe decomposition, where both ship schedules and inventory decisions can be expressed as columns. Christiansen and Fagerholt (2002) and Christiansen and Nygreen (2005) have later extended the deterministic problem by allowing soft inventory and time windows constraints, as a way to overcome uncertainty and stochasticity which are intrinsic to the problem. For a comprehensive review of ship routing and scheduling, the reader is referred to Christiansen et al. (2004) and to Christiansen et al. (2007).

Another version of the IRP is concerned with split deliveries, that is, the demand of any given customer can be satisfied by more than one vehicle. This increases the flexibility of the system and may lead to reduced transportation costs. Chandra and Fisher (1994) proposed a multi-period multi-product problem with split deliveries. Their two phase heuristic solution procedure was proved to generate infeasible solutions, and only yielded lower bounds to the problem (Yu et al., 2007). Fumero and Vercellis (1999) also worked on split deliveries for a production, inventory and routing multi-period problem for a single item, and solved it by using Lagrangian relaxation. Yu et al. (2008) improved previous works on split deliveries by adding extra valid inequalities to reduce the solution space and by developing a more robust Lagrangian relaxation approach. The relaxed problem can be approximately solved, and their solutions are used to construct feasible routes from the original problem.

Policy based heuristics are also present in the literature. Power-of-two policies are analyzed by Herer and Roundy (1997); a fixed partition policy combined with a tabu search heuristic is studied by Zhao et al. (2007); for a multi-product problem, Viswanathan and Mathur (1997) have developed a stationary nested joint replenishment policy. A review of matheuristics for the IRP is provided in Bertazzi and Speranza (2012).

Some real-life features are added in the study presented in Benoist et al. (2011), such as multi-plants, resources that must be combined to make a route (driver, trailer, tractor), driver working hours constraints, and two kinds of resupply (order-based and forecast). The problem was modeled with a surrogate objective function to take into account the long-term cost, since the problem is solved over a rolling horizon using simple moves on a randomized local search.

Table 2.4 presents the main papers cited in this section. Tables 2.5 and 2.6 summarize some of the features just discussed for the deterministic and stochastic IRP, respectively.

As can be seen, previous research have given little attention to flexibility and none to consistency, which are the main themes of this thesis. However, there have been some work on transshipment outside the IRP framework, which can benefit the IRP. Here we review the relevant transshipment literature in Section 2.6. Existing research on consistency issues have been mostly conducted in the context of the VRP and will be mentioned in Chapter 4.

2.6 Transshipments in inventory management

Although there does not appear to exist any literature related to transshipments in an IRP context, there exists some research about it in inventory management, as a means of sharing inventory among several locations, also sharing the risks of stockout.

Transshipments are movements of goods through an intermediate location, before the shipment to the final destination. In supply chain management, this possibility often arises when there is a need to change the means of transportation or to combine small loads into a larger one. However, transshipment can also be a valuable option in the following situation. Let a company have several retailers with on-site inventory, all served by a central warehouse as we have assured so far for the IRP. If retailer A faces a high demand and low inventory situation, while retailer B still has enough inventory, it may be interesting to ship goods from B to A, without the need to dispatch a vehicle to serve A from the warehouse, possibly incurring an emergency cost. The assumption is that transshipments will reduce total cost for the whole system (considering that the supplier and the retailers belong to the same chain), while increasing service level (Tagaras, 1999). Thus, transshipments will be treated in this section as an operational solution towards cost reduction, with possible applications to the IRP, as opposed to the tactical approach given by Herer et al. (2002) who treat transshipments as a solution leading to increased supply chain agility. Different definitions of transshipments exist, such as the one used by Dondo et al. (2009) who studies a pickup and delivery problem in which good can be transferred from one vehicle to another. The same problem is also studied by Qu and Bard (2012) who proposes a GRASP with adaptive large neighborhood search for its solution.

The literature can usually be categorized according to the following criteria. The list is not extensive as those are the categories believed to be the most important for

Table 2.4: Classification of main papers on other versions of the IRP

Authors	**Year**	**Time horizon**		**Demand**		**Products**			**Structure**			**Routing**		**Inventory policy**		**Fleet composition**		**Fleet size**		
		Finite	Infinite	Deterministic	Stochastic	Single	Two	Many	One-to-one	One-to-many	Many-to-many	Direct	Multiple	Lost sales	Backlogging	Homogeneous	Heterogeneous	Single	Multiple	Unconstrained
Blumenfeld et al.	1985		✓	✓				✓		✓	✓	✓		✓		✓				✓
Federgruen et al.	1986	✓			✓		✓			✓			✓	✓			✓		✓	
Speranza and Ukovich	1994		✓	✓				✓		✓		✓		✓		✓				✓
Carter et al.	1996	✓		✓				✓		✓			✓		✓	✓			✓	
Bertazzi et al.	1997		✓	✓				✓		✓			✓	✓		✓				✓
Qu et al.	1999		✓		✓			✓		✓			✓	✓		✓		✓		
Bertazzi and Speranza	2002	✓		✓				✓	✓			✓		✓		✓			✓	
Bertazzi et al.	2002	✓		✓				✓		✓			✓	✓		✓		✓		
Adelman	2003		✓	✓		✓	✓	✓		✓			✓	✓		✓			✓	
Persson and Göthe-Lundgren	2005	✓		✓				✓			✓		✓	✓			✓		✓	
Stacey et al.	2007		✓	✓				✓		✓			✓	✓		✓			✓	
Christiansen et al.	2011		✓	✓				✓			✓		✓	✓			✓		✓	

Table 2.5: Characteristics of main papers addressing the deterministic IRP

	Dror et al. (1985)	Blumenfeld et al. (1985)	Dror and Ball (1987)	Chien et al. (1989)	Anily and Federgruen (1990)	Gallego and Simchi-Levi (1990)	Speranza and Ukovich (1994)	Carter et al. (1996)	Bertazzi et al. (1997)	Christiansen (1999)	Bertazzi and Speranza (2002)	Bertazzi et al. (2002)	Adelman (2003)	Campbell and Savelsbergh (2004)	Persson and Göthe-Lundgren (2005)	Aghezzaf et al. (2006)	Archetti et al. (2007)	Stacey et al. (2007)	Savelsbergh and Song (2008)	Abdelmaguid et al. (2009)	Raa and Aghezzaf (2009)	Archetti et al. (2012)	Aksen et al. (2012)
Finite	✓		✓	✓			✓	✓		✓	✓	✓		✓	✓		✓		✓	✓	✓	✓	
Infinite		✓			✓	✓			✓				✓			✓		✓			✓		
One item	✓		✓	✓	✓	✓				✓			✓	✓		✓	✓		✓	✓	✓	✓	✓
Multi items		✓					✓	✓	✓		✓	✓	✓		✓			✓					
One-to-one			✓								✓												
One-to-many	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓		✓	✓	✓		✓	✓	✓	✓
Many-to-many		✓								✓					✓				✓				
Direct deliveries		✓				✓	✓				✓												
Multiple deliveries	✓		✓	✓	✓			✓	✓			✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Continuous delivies										✓									✓				
Backlogging				✓				✓												✓			
Homogeneous fleet	✓	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓
Heteregeneous fleet				✓						✓					✓					✓			
Single vehicle												✓					✓					✓	✓
Multiple vehicles	✓			✓	✓			✓		✓	✓		✓	✓	✓	✓		✓	✓	✓	✓		
Unconstrained vehicles		✓			✓	✓	✓		✓														

Table 2.6: Characteristics of main papers addressing the stochastic IRP

	Bell et al. (1983)	Federgruen and Zipkin (1984)	Golden et al. (1984)	Federgruen et al. (1986)	Dror and Ball (1987)	Trudeau and Dror (1992)	Minkoff (1993)	Bard et al. (1998)	Qu et al. (1999)	Berman and Larson (2001)	Kleywegt et al. (2002)	Adelman (2004)	Kleywegt et al. (2004)	Hvattum and Løkketangen (2009)	Hvattum et al. (2009)	Bertazzi et al. (2012)
Finite	✓	✓	✓	✓	✓	✓		✓		✓						
Infinite							✓		✓	✓	✓	✓	✓	✓	✓	
One item	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
Multi items				✓					✓							
One-to-one					✓	✓										
One-to-many	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Many-to-many								✓								
Direct deliveries											✓					
Multiple deliveries	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
Continuous delivies																
Backlogging																
Homogeneous fleet			✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Heteregeneous fleet	✓	✓		✓												
Single vehicle									✓	✓		✓				
Multiple vehicles	✓	✓	✓	✓	✓	✓		✓				✓	✓	✓	✓	
Unconstrained vehicles							✓				✓					

the IRP:

- number of locations that are able to ship goods: since the problem is complex from the mathematical point of view, the number of locations operating as intermediate inventories is relevant;
- the lead time (Tagaras, 1989; Tagaras and Cohen, 1992) and cost (Robinson, 1990) for a replenishment from the warehouse, compared to those same parameters when using the transshipment option;
- whether the transshipment has a preventive (Das, 1975; Diks and de Kok, 1996; Jönsson and Silver, 1987; Mercer and Tao, 1996) or emergency (Cohen et al., 1986; Lee, 1987; Axsäter, 1990; Dada, 1992; Tagaras, 1999; Suakkaphong and Dror, 2011) purpose, and so, if it is decided before or after demand is observed;
- the measure of performance: cost or service level (Herer et al., 2002).

Unless specified, all papers mentioned below consider that the decisions are centralized by a "parent firm", instead of being local decisions. This is important because the parent firm would be concerned about the overall system performance, instead of optimizing each part of it.

The transshipment literature dates back to the 1950s when Allen (1958) developed a model to redistribute inventory at the beginning of each period to each of the n locations facing normally distributed demands. The decisions are centrally coordinated, and any shortage during the period is lost. Gross (1963) develops this model a little further with the possibility to buy additional units of the goods and by generalizing the demand at the retailers to any distribution, instead of only the normal one as done by Allen (1958). However, in these two models all parameters are the same for all retailers. Karmarkar and Patel (1977) generalize this assumption allowing retailer-specific values.

Krishnan and Rao (1965) have analyzed the case where locations are identical both in cost and demand, whereas Robinson (1990) have later extended this approach to a two-location model with non-identical costs, introducing a linear programming based heuristic for the multi-location model. According to Nonås and Jörnsten (2007), this is a general rule found in literature: analytical results for the two-location model (e.g. Tagaras (1999); Rudi et al. (2001)) or heuristics methods for the n location model (e.g. Karmarkar and Patel (1977); Robinson (1990)).

For the two-location model, Tagaras (1999) derives a set of assumptions which make the full transshipments, called complete pooling, an optimal policy. Complete pooling assumes that the retailers are willing to transfer any excess inventory to others with a shortage until either the shortage or the excess is eliminated. Herer and Rashit (1999) have also studied the two-location problem but with special cost characteristics: in their model, there are a fixed and a joint replenishment costs. Finally, Chen et al. (2012) show that when lead times are very long and the supplier has only one chance to supply goods to its two retailers, i.e. at the beginning of the selling season, there exists a unique optimal transshipment policy for the subsequent periods. Moreover, transshipments are shown to increase both supplier and retailers' expected profit as well as retailers' service levels.

Rudi et al. (2001) study the two-location model where there is local decision making, that is, each retailer is interested in maximizing its own profit, since he is not part of a corporation trying to maximize the whole system profit. For the retailer alone, the possibility of transshipments affects the way he will manage his inventory, since there exists the possibility of holding lower inventory levels while still benefiting from the safety offered by transshipments if ever the demand turns out to be higher than expected. In Suakkaphong and Dror (2011) it is even shown that it can be benefical to transship inventory regardless of local demand, if this increases the overall profit of the system. The case in which customers sharing their inventory with each other through transshipments do not belong to the same chain, but rather compete for customers is further analyzed by Zou et al. (2010). These authors show that transshipments are not a viable option if companies are close competitors, allowing customers to easily switch to one another in case of stockouts. Satır et al. (2012) also study this system configuration in an decentralized service parts network, and analyze the effects of different levels of inventory sharing as well as that of information sharing. Under the same framework in which retailers are competitors, Mateo et al. (2012) propose a game theory methodology which allows each retailer to sell surplus inventory to other retailers. This is shown to increase one's own profit margin while cooperating with one's competitor.

In the literature devoted to heuristics, Robinson (1990), besides solving to optimality the problem with two customers (two location problem), approximates the optimal solution for the n location model by discretizing the demand distribution and solving a large linear programming problem heuristically. Tayur (1995) also discretizes demand distribution for the n location model, but his gradient approach is faster and the size of the resulting problem is smaller. Herer et al. (2006) used Monte Carlo simulation in order to derive an order-up-to S policy for the non-identical multi location transshipment problem.

Wee and Dada (2005) review several models that specify where the goods should be shipped from, if one of the following scenarios occur:

- if there is no central depot and all transshipments should start from a retailer.
- if there is a warehouse, but transshipments should start from a retailer, if some has excess inventory.
- if there is a warehouse, and transshipments should start from it, unless it has no inventory.
- if the transshipment is not allowed to start from a retailer, only from the warehouse.
- and finally if it is not allowed to perform transshipments in the system.

The exact formulation described below was initially proposed by Nonås and Jörnsten (2005) and further developed in Nonås and Jörnsten (2007) to include a penalty in case the final customer demand cannot be satisfied. We will describe the most general one.

The description and notation used is the following. Consider n retailers belonging to the same corporation selling a seasonal product. They have to determine, before the season starts and before knowing the actual demand D_i of each store i, a quantity Q_i to keep in inventory for the coming season. Demands are supposed to be stochastic and unknown, but their probability distributions are supposed to be known and continuous.

For each unit store i sells, it receives a revenue r_i, strictly greater than the unit ordering cost c_i. If after the season ends, store i still has units in inventory, it can sell them back to the factory or put them on sale for a salvage value s_i, which is smaller than the ordering cost c_i. Also, if store j faces a shortage during the season, it is possible to ship goods to it from another store i having a surplus of the product, at a unit cost τ_{ij}, and it is assumed that the transportation time is negligible. Let T_{ij} represent the amount of products sold at location j from the inventory at location i. Naturally, T_{ii} is the amount sold at location i from its own inventory and τ_{ii} is set to zero. If there is a shortage at location i and there is no location with surplus inventory to transship from, then location i incurs a penalty cost p_i.

In order to formulate a model with complete pooling, three assumptions have to be made on some of the above described parameters. These assumptions are common in real life applications and similar ones can be found in the literature (Tagaras, 1999; Herer and Rashit, 1999; Robinson, 1990). They are the following:

$$r_j - \tau_{ij} \geq s_i \qquad i, j = 1, \ldots, n; \tag{2.41}$$

$$r_i \geq r_j - \tau_{ij} \qquad i,j = 1,\ldots,n; \tag{2.42}$$

$$s_i \geq s_j - \tau_{ij} \qquad i,j = 1,\ldots,n; \tag{2.43}$$

$$c_i + \tau_{ij} \geq c_j \qquad i,j = 1,\ldots,n. \tag{2.44}$$

Inequality (2.41) implies that it is always better to transship excess goods to a location with a shortage than to sell them at the salvage cost. Constraints (2.42) and (2.43) mean that it is never optimal to transship between two shortage locations or between two surplus locations, respectively. Finally, it is better to order from the factory than to order from any other location, as stated by inequality (2.44).

Still according to Nonås and Jörnsten (2007), if all retailers belong to same company, it would be optimal to maximize the total aggregated profit instead of individual ones. This is made possible when decisions are centrally coordinated, and the maximum aggregate profit for all n locations is

$$\text{maximize } \pi = \max\left\{\sum_{i=1}^{n} -c_i Q_i + E\bar{K}[Q,D]\right\}, \tag{2.45}$$

where $\bar{K}$ is the maximum income given order quantities Q and realized demands D. Since the complete pooling is the policy being used, all units transshipped are sold at the receiving location. Thus,

$$\bar{K}[Q,D]{=}\text{maximize} \sum_{i=1}^{n}\left[\sum_{j=1}^{n} r_j T_{ij} - \sum_{j=1}^{n} \tau_{ij} T_{ij} + s_i\left(Q_i - \sum_{j=1}^{n} T_i j\right) - p_i\left(D_i - \sum_{j=1}^{n} T_{ij}\right)\right] \tag{2.46}$$

subject to

$$\sum_{j=1}^{n} T_{ij} \leq Q_i \qquad i = 1,\ldots,n; \tag{2.47}$$

$$\sum_{j=1}^{n} T_{ij} \leq D_i \qquad i = 1,\ldots,n; \tag{2.48}$$

$$T_{ij} \geq 0 \qquad i,j = 1,\ldots,n. \tag{2.49}$$

Equation (2.46) defines the maximum income, and its right-hand side is the income derived from all products sold at locations j coming from the inventory at location i. Again, if i and j are the same, then it refers to its own inventory. It is made up of the selling revenue, minus transshipments costs, plus salvage costs, minus any eventual penalty costs. Constraints (2.47) make it impossible to sell more than the inventory, and constraints (2.48) state that one cannot sell more than its demand. Constraints (2.49) are simply non-negativity constraints.

Adapting equations (2.45) and (2.46), by extracting $\sum_{i}^{n} s_i Q_i$ and $\sum_{i}^{n} p_i D_i$ from $\bar{K}$, makes it possible to reformulate the model as:

$$\text{maximize } \pi = \max\left\{-\sum_{i=1}^{n}\left[\left(c_i - s_i\right) Q_i\right] + p_i E[D_i] + EK[Q,D]\right\} \tag{2.50}$$

where

$$K[Q,D] = \max \sum_{i=1}^{n} \sum_{i=1}^{n} (r_j + p_j - \tau_{ij} - s_i)\, T_{ij} \qquad (2.51)$$

subject to

$$\sum_{j=1}^{n} T_{ij} \leq Q_i \qquad i = 1, \ldots, n; \qquad (2.52)$$

$$\sum_{j=1}^{n} T_{ij} \leq D_i \qquad i = 1, \ldots, n; \qquad (2.53)$$

$$T_{ij} \geq 0 \qquad i, j = 1, \ldots, n. \qquad (2.54)$$

The transshipment problem can be solved heuristically by discretizing the continuous demand distribution. Several authors use this procedure (Robinson, 1990; Tayur, 1995; Nonås and Jörnsten, 2005, 2007). Tayur (1995) showed that there is no minimal number of discrete demand points that can guarantee the solution to be within a given error bound of the optimal one. Nonetheless, Nonås and Jörnsten (2005) compared the accuracy of several discretization scenarios against the one with 30,000 points, and concluded that there is a significant improvement when the discretization moves from 100 points as compared to 30,000, and also from 1,000 to 30,000. However, when using a discrete distribution with 10,000 points, the average improvement when using 30,000 is never above 0.5%.

The transshipment problem was also recently solved exactly using stochastic programming by Gong and Yücesan (2012) for problems with negligible transshipment lead times, stochastic demand and a specific inventory policy.

2.7 Consistency in vehicle routing

Consistency in the context of the IRP does not seem to have been studied in the manner we propose in this thesis. Here, the OU policy, already studied in the IRP (see e.g. Bertazzi et al. (2002)), is viewed as a consistency feature (see Chapter 4). However, traditionally it has been imposed as a way to simplify the problem, linking the decisions regarding when to deliver and how much to deliver to a customer into only one, in order to make the problem more computationally tractable.

However, consistency as we define it has arisen in the context of the VRP in the past. The concept of driver consistency was first introduced by the work of Groër et al. (2009). It was studied in the context of the periodic VRP (PVRP), in which besides the constraints of a CVRP one considers that a set of customers have to be visited, one or several times, over a given planning horizon. A list of possible sets of visiting days for each customer is given. In the consistent PVRP, one must ensure that each customer is always visited by the same driver, so as to develop a personal rapport that over time should lead to a better relationship between the customer and the driver. We call this feature driver consistency, as different types of consistency will be proposed and analyzed in Chapter 4. Moreover, the vehicle operation would benefit from the driver's increased familiarity with the region and the customer site. According to the authors, imposing such constraints yields solutions having a higher cost and requiring slightly more vehicles than in the inconsistent solutions. Obviously, all consistent solutions are feasible for the more general problem.

There also exist papers that incorporate workforce management within the PVRP, for example by assigning territories to drivers as Christofides and Beasley (1984), Beasley (1984) and Zhong et al. (2007). This is an indirect way to enforce driver consistency. Recently, Smilowitz et al. (2012) have analyzed the trade-off between workforce management and travel distance goals in a multi-objective PVRP.

2.8 Discussion and future work

Given the diversity of versions of the IRP, it is difficult to develop a proper literature review analyzing evolutions and comparing methods used to assess and solve the problem over the years. Several approaches have been developed for the IRP, none outperforming the others in a manner that it would be preferred by most researchers. Moreover, there still seems to be room for different methods. Added to this, there does not yet exist a set of instances universally used to benchmark the proposed algorithms, making it difficult to compare their effectiveness and efficiency.

Due to the intrinsic difficulty of the problem, exact methods are barely used, and one can find several heuristics to tackle different IRPs. However, one cannot find consensus about what to solve optimally and what to solve heuristically. For instance Bell et al. (1983) used heuristics to generate forecasts and possible routes, Dror et al. (1985) and Dror and Ball (1987) used them to choose replenishment periods, Anily and Federgruen (1990) to determine the set of customers to be visited, and so on.

Despite these differences, most papers have one point in common: they consider that the problem is solved only once (i.e. it is static) with known (or deterministic) data. However, there exist several situations where the problem must be solved in a dynamic fashion, i.e. as new information is revealed. In such contexts, one may wish to make use of forecasts (typically of demands) based on statistical information. Assessing the expected value of information is an interesting research question in a dynamic setting. An emerging solution paradigm proposed to deal with uncertainty is robust optimization which constructs an a priori solution under an uncertainty budget in order to yield a solution capable of withstanding variants in inputs (Solyalı et al., 2012).

Other practical issues usually involve omitted constraints, such as time windows to perform deliveries at the customer sites, drivers' working hours, a possible heterogeneous fleet of vehicles, multiple source problems (when the supplier has more than one warehouse), multi-product, and of course, the integration of all these constraints within one solvable model. Furthermore, it is not trivial to obtain parameters such as inventory holding costs and penalties, since some of them are hard to measure and sometime intangible. These parameters are essential to model real-life applications, and solutions can drastically change as a result of variations in their values.

The VMI is also concerned with the relationship of supplier and retailer, but one may want to consider the integration of production, storage and transportation, managing inventory costs at all three levels, and integrating them into that all transportation and production costs.

The use of transshipments can lead to added flexibility but to increased computational difficulty. To the best extent of the knowledge gained in this research, one cannot find references in the literature to the combination of IRP and transshipments. The resulting problem is likely to be huge and unsolvable exactly, but heuristics may be successful. Likewise, quality of service features can probably be introduced in the IRP through the use of consistency attributes. Despite the existence of some research about the consistent VRP, its integration within the IRP has yet to be developed.

This chapter has provided a review of the IRP as part of the overall logistics management system. The integration of the inventory and routing aspects is proved to reduce costs, especially when it is coordinated at the various echelons of the supply chain, possibly through a VMI system.

Some researchers have attempted to solve the problem exactly as seen in Section 2.5.1. Nevertheless, there is still a long way to go in order to put into practice most of the knowledge developed, especially because one cannot compare the algorithms each study proposes, and one does not yet know the best way to solve each version of the problem. This is due to the fact that it is only recently that a set of benchmark instances has been introduced and shared among researchers.

More coordination and cooperation than what is usually found in VMI is possible. This can possibly be achieved through transshipments, as well as inventory sharing and risk sharing. This integration should be addressed and tested to confirm or refute this hypothesis, taking into account not only cost parameters but also the efficiency of the proposed solution methods, since it is unlikely that an exact algorithm can be developed to solve the IRP with transshipments in the near future.

Nevertheless, transshipment and consistency have not yet been integrated within the IRP. In a deterministic context, transshipments can still reduce distribution costs and consistency can be used as a means of improving the quality of service. In a dynamic and stochastic environment, they can act as a way to reduce stockout risks, yet offering good consistent decisions over time. These are the main topics that we aim to introduce and analyze in this thesis.

Chapter 3

The Inventory-Routing Problem with Transshipment

Chapter information

An article based on this chapter was published in *Computers & Operations Research*: L. C. Coelho, J.-F. Cordeau, G. Laporte. The Inventory-Routing Problem with Transshipment. *Computers & Operations Research*, 39(11):2537−2548, 2012.

An article partly based on the exact algorithm presented in Section 4.3 was published in *Computers & Operations Research*: L. C. Coelho, G. Laporte. Exact Solutions for Several Classes of Inventory-Routing Problems. *Computers & Operations Research*, 40(2):558−565, 2013.

In this chapter we analyze how IRP solutions can be made more flexible. To this end, we introduce the concept of *transshipment* within inventory-routing.

3.1 Introduction

In addition to the features of the IRP described in Chapter 2, which makes possible the application of a VMI strategy, this chapter introduces the concept of *transshipment* within inventory-routing. Under this policy, goods may be shipped to a customer, either directly from the supplier, or from another customer. This happens, for example, between stores belonging to the same chain which can ship merchandise to one another when unforeseen demands variations occur (Axsäter, 1990; Dada, 1992; Lee, 1987; Nonås and Jörnsten, 2005, 2007; Paterson et al., 2011).

To the best of our knowledge, transshipment has not yet been formally integrated within the context of inventory-routing. Planned transshipments can also be used to redistribute inventory among customers so as to reduce handling costs, as is the case in the retail industry (Paterson et al., 2011) and in companies that make use of external freight carriers for part of their distribution (Nonås and Jörnsten, 2007). Transshipments may be beneficial in a deterministic context in which no shortages occur because they may yield an overall reduced distribution and inventory holding cost. This is the case, for example, when vehicle capacity and storage limits at customer locations restrict the amounts that can be delivered to these customers at each time period. Deterministic subproblems also arise when solving stochastic inventory-routing problems in a rolling horizon framework where one uses demand forecasts for the next time periods as approximations of the unknown demand. This is the context in which our problem is defined. Mercer and Tao (1996) provide an example of an inventory-routing system used by the supermarket chain Tesco, in the United Kingdom, where deliveries are made from a factory to several warehouses, and lateral transshipments can take place between warehouses. Note that the concept of transshipment also appears in a different way in Shen et al. (2011). The latter paper describes a three-level supply chain consisting of suppliers, transshipment ports and customers, but no transshipment takes place among customers.

As pointed out by Laporte (2009), relatively medium-size instances of the VRP cannot be solved exactly using exact methods. Incorporating inventory management issues and transshipments makes the problem significantly harder. We have developed a branch-and-cut algorithm to evaluate the problem exactly and an adaptive large neighborhood search (ALNS) heuristic for it. The latter was initially put forward by Ropke and Pisinger (2006a) in the context of the VRP and extends a concept initially proposed by Shaw (1997). The algorithm we propose is designed to handle the specific features of the IRPT. It is flexible and can easily handle the OU and ML replenishment policies.

The main scientific contributions of this chapter are the introduction of a transshipment option within the context of inventory-routing, the development of an exact algorithm to solve it and the development of a powerful and flexible ALNS heuristic to solve four variants of the problem: the IRPT with transshipment (IRPT) and the IRP without transshipment (IRP), under an OU or an ML replenishment policy. More specifically, we show that ALNS provides a powerful algorithmic framework capable of simultaneously handling the routing, scheduling and transshipment decisions inherent to the IRPT. In addition, we demonstrate on benchmark instances the advantage of allowing transshipments. The four variants will be referred to as

IRPT-OU, IRPT-ML, IRP-OU and IRP-ML.

The remainder of the chapter is organized as follows. In Section 3.2 we introduce and describe the IRP. Section 3.3 presents two mixed-integer linear programming formulations for the four variants of the problem considered in the chapter, and for a restriction in which routing is fixed, followed by a branch-and-cut algorithm in Section 3.4. Our ALNS algorithm is presented in Section 3.5, followed by computational results, in Section 3.6, and by our conclusions in Section 3.7.

3.2 Problem description

We now formally introduce the IRPT. The problem is defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ where $\mathcal{V} = \{0, ..., n\}$ is the vertex set and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ is the arc set. Vertex 0 represents the supplier and the vertices of $\mathcal{V}' = \mathcal{V} \setminus \{0\}$ represent customers. Both the supplier and customers incur unit inventory holding costs h_i per period ($i \in \mathcal{V}$), and each customer has an inventory holding capacity C_i. The length of the planning horizon is p and, at each time period $t \in \mathcal{T} = \{1, ..., p\}$, the quantity of product made available at the supplier is r^t. We assume the supplier has enough inventory to meet all the demand during the planning horizon and that inventories are not allowed to be negative, i.e., the supplier can only ship what he holds in stock with no backlogging option. At the beginning of the planning horizon the decision maker knows the current inventory level of the supplier and of the customers (I_0^0 and I_i^0), and receives the information on the demand d_i^t of each customer i for each time period t. Throughout the paper, we assume that the quantity r^t becoming available at the supplier in period t can be used for deliveries to customers in the same period, and that the quantities q_i^t received by customer i in period t can be used to meet the demand in that period.

A single vehicle of capacity Q is available. This vehicle is able to perform one route at the beginning of each time period to deliver products from the supplier to a subset of customers. A routing cost c_{ij} is associated with arc $(i, j) \in \mathcal{A}$. Whereas many distribution systems make use of several vehicles, most research in the field of inventory-routing still considers only one vehicle, and there are indeed practical applications in which a single vehicle is used at a given echelon of the supply chain, such as in the case study described by Mercer and Tao (1996).

Transshipments can be made later in the time period. A transshipment can start from any customer in a subset $\mathcal{R} \subseteq \mathcal{V}'$, i.e., these customers can dispatch goods to other customers as needed. Direct deliveries from the depot are also allowed. Transshipments and direct deliveries can occur when it is profitable to ship goods from the depot to a customer on a special request basis, or from customer $i \in \mathcal{R}$ to customer $j \in \mathcal{V}'$. This can be done by subcontracting to a carrier who will pickup goods either at the supplier or from any transshipment point. These outsourced deliveries are only made by direct shipping and the unit cost associated with transshipping products from i to j is b_{ij}. For the sake of simplicity, throughout this chapter we will use the word *transshipment* indiscriminately when refering both to lateral transshipments and direct deliveries.

It is possible that both the supplier's vehicle and the subcontractor visit the same customer within the same time period: the supplier's vehicle first delivers to the customer according to the OU or to the ML policy, and the subcontractor may later deliver to that customer according to the ML policy. The total quantity delivered to a customer in a given period guarantees that no shortages occur and that the capacity is not exceeded at the end of the period. However, the customer's capacity may be temporarily exceeded during that period. We also assume that all orders and deliveries can be performed during the same time period, which means that lead times are negligible.

The objective of the problem is to minimize the total cost while meeting the demand for each customer in each period. The replenishment plan is subject to the following constraints:

- the inventory level of a customer at the end of a period cannot exceed the maximal available inventory capacity;
- inventories are not allowed to be negative, i.e., all demand must be met by previous inventory plus deliveries performed during the time period considered;
- if the supplier's vehicle visits a customer in a time period, an OU or an ML replenishment policy applies;
- the supplier's vehicle can perform at most one route per time period, starting and ending at the supplier;
- the vehicle capacity cannot be exceeded.

The solution to the problem should determine (1) which customers to serve in each time period using the supplier's vehicle, (2) which route to use in each time period, and (3) how much to transship from every $i \in \mathcal{R} \cup \{0\}$ to every $j \in \mathcal{V}'$ in each time period. We assume that the following sequence of events takes place:

- At the start of the planning horizon the quantities I_i^0 $(i \in \mathcal{V})$ and d_i^t $(i \in \mathcal{V}', t \in \mathcal{T})$ are known.

- At every period $t \in \mathcal{T}$ routes are performed and quantities q_i^t are delivered, transshipments w_{ij}^t take place, demands d_i^t occur, and inventory levels I_i^t are measured.

3.3 Mathematical models

We now introduce directed and undirected models for several versions of the IRPT.

3.3.1 Directed model for the IRPT-OU

The model works with the following binary variables: x_{ij}^t is equal to 1 if and only if customer j immediately follows customer i on the route of the supplier's vehicle in period t. Let w_{ij}^t be the amount of product delivered directly from $i \in \mathcal{R} \cup \{0\}$ to customer $j \in \mathcal{V}'$ at period t using the outsourced carrier. Let I_i^t denote the inventory level at vertex $i \in \mathcal{V}$ at the end of period $t \in \mathcal{T}$. We denote by q_i^t the quantity of product delivered from the supplier to customer i in time period t. The model also uses continuous variables v_i^t to enforce the VRP subtour elimination contraints (Desrochers and Laporte, 1991; Kara et al., 2004). They represent the sum of the deliveries made by the vehicle in period t after visiting customer i.

In the IRPT, the total cost to be minimized is the sum of inventory holding costs at the supplier and at the customers, of routing costs for the supplier's vehicle and of transshipment costs:

$$\min \sum_{t \in \mathcal{T}} h_0 I_0^t + \sum_{i \in \mathcal{V}'} \sum_{t \in \mathcal{T}} h_i I_i^t + \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} \sum_{t \in \mathcal{T}} c_{ij} x_{ij}^t + \sum_{i \in \mathcal{R} \cup \{0\}} \sum_{j \in \mathcal{V}'} \sum_{t \in \mathcal{T}} b_{ij} w_{ij}^t. \qquad (3.1)$$

As is standard in vehicle routing, travel costs are distance-dependent and are unrelated to the vehicle load. However, transshipment costs are distance- and volume-dependent because this is sometimes how outsourced carriers define the terms of their contracts.

The constraints are as follows.

3.3.1.1 Inventory definition at the supplier

The inventory level at the supplier at the end of period $t \in \mathcal{T}$ is given by the inventory level at the end of period $t-1$, plus the quantity r^t made available in period t, minus the total quantity shipped to the customers using the supplier's vehicle in period t, minus the total quantity transshipped to the customers in period t:

$$I_0^t = I_0^{t-1} + r^t - \sum_{i \in \mathcal{V}'} q_i^t - \sum_{i \in \mathcal{V}'} w_{0i}^t \quad t \in \mathcal{T}. \tag{3.2}$$

3.3.1.2 Stockout constraints at the supplier

These constraints impose that the supplier's inventory at the end of period t cannot be negative:

$$I_0^t \geq 0 \quad t \in \mathcal{T}. \tag{3.3}$$

3.3.1.3 Inventory definition at the customers

Likewise, the inventory level at each retailer in period t is given by its previous inventory level in period $t-1$, plus the quantity q_i^t delivered by the supplier's vehicle in period t, plus the total quantity transshipped in period t, minus the total quantity transshipped to other customers in period t, minus its demand in period t, that is:

$$I_i^t = I_i^{t-1} + q_i^t + \sum_{j \in \mathcal{R} \cup \{0\}} w_{ji}^t - \sum_{j \in \mathcal{V}'} w_{ij}^t - d_i^t \quad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{3.4}$$

3.3.1.4 Transshipment origins

The transshipment quantities w_{ij}^t must be set to zero if i is not in the set $\mathcal{R}$:

$$\sum_{j \in \mathcal{V}'} w_{ij}^t = 0 \quad i \notin \mathcal{R} \quad t \in \mathcal{T}. \tag{3.5}$$

3.3.1.5 Stockout constraints at the customers

These constraints guarantee that for each customer $i \in \mathcal{V}'$ the inventory level I_i^t remains non-negative at all time:

$$I_i^t \geq 0 \quad i \in \mathcal{V} \quad t \in \mathcal{T}. \tag{3.6}$$

3.3.1.6 Maximal inventory level at the customers

These constraints guarantee that for each customer $i \in \mathcal{V}'$ the inventory level I_i^t remains below the maximum level C_i at the end of each period:

$$I_i^t \leq C_i \quad i \in \mathcal{V} \quad t \in \mathcal{T}. \tag{3.7}$$

3.3.1.7 Quantities delivered

These sets of constraints ensure that the quantity delivered by the supplier's vehicle to each customer $i \in \mathcal{V}'$ in each period $t \in \mathcal{T}$ will fill the customer's inventory capacity if the customer is served, and will be zero otherwise:

$$q_i^t \geq C_i \sum_{j \in \mathcal{V}'} x_{ij}^t - I_i^{t-1} \quad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{3.8}$$

$$q_i^t \leq C_i - I_i^{t-1} \quad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{3.9}$$

$$q_i^t \leq C_i \sum_{j \in \mathcal{V}} x_{ij}^t \quad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{3.10}$$

If customer i is not visited in period t, then constraints (3.10) mean that the quantity delivered to it will be zero (while constraints (3.8) and (3.9) are still respected). If, otherwise, customer i is visited in period t, constraints (3.10) limit the quantity delivered to the customer's inventory holding capacity, and this bound is tightened by constraints (3.9), making it impossible to deliver more than what would exceed this capacity. Constraints (3.8) model the OU replenishment policy, ensuring that the quantity delivered will be exactly the bound provided by constraints (3.9).

3.3.1.8 Vehicle capacity

These constraints guarantee that the vehicle's capacity is not exceeded:

$$\sum_{i \in \mathcal{V}'} q_i^t \leq Q \quad t \in \mathcal{T}. \tag{3.11}$$

3.3.1.9 Routing constraints

These constraints guarantee that a feasible route is designed to visit all customers served in period t:

a) Flow conservation constraints: these constraints impose that the number of arcs entering and leaving a vertex should be the same:

$$\sum_{i \in \mathcal{V}} x_{ij}^t = \sum_{i \in \mathcal{V}} x_{ji}^t \quad j \in \mathcal{V} \quad t \in \mathcal{T}. \tag{3.12}$$

b) A single vehicle is available:

$$\sum_{i \in \mathcal{V}} x_{i0}^t \leq 1 \quad t \in \mathcal{T}. \tag{3.13}$$

c) Subtour elimination constraints:

$$v_i^t - v_j^t + Qx_{ij}^t \leq Q - q_j^t \quad i \in \mathcal{V}' \quad j \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{3.14}$$

$$q_i^t \leq v_i^t \leq Q \quad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{3.15}$$

3.3.1.10 Integrality and nonnegativity constraints

$$v_i^t, q_i^t, w_{ji}^t \geq 0 \quad i \in \mathcal{V}' \quad j \in \mathcal{R} \cup \{0\} \quad t \in \mathcal{T}; \tag{3.16}$$

$$x_{ij}^t \in \{0, 1\} \quad i, j \in \mathcal{V}, i \neq j \quad t \in \mathcal{T}. \tag{3.17}$$

3.3.2 Adaptations to IRPT-ML, IRP-OU and IRP-ML

The IRPT-OU model just described can be modified to enforce the ML replenishment policy by dropping constraints (3.8). Similarly, to forbid transshipments one only has to set all w_{ij}^t variables equal to zero. Thus, all four versions of the IRPT can be modeled through the same formulation.

3.3.3 Network flow model for the IRPT with fixed routes

If one fixes routing variables x_{ij}^t, the remaining problem reduces to a network flow problem defined by the I_i^t, q_i^t and w_{ij}^t variables. The flow conservation equations are given by (3.2) and (3.4). The lower and upper bounds on the flows are defined by (3.3) and (3.6)−(3.10). Vehicle capacity constraints (3.11) still define an upper bound on the quantity delivered by the vehicle, even though the customers to be visited are fixed. Constraints (3.12)−(3.14) are not relevant in the flow models because their variables are fixed.

Figure 3.1 depicts the network flow model for a small network with two customers and two time periods. The supplier and the customers are represented by vertices replicated for each time period, plus one extra set of vertices for initial inventories, and one extra set for the decisions made at the last time period. The supplier and each customer carry their inventories between successive time periods. The corresponding solid arcs in the figure have unit inventory holding costs, and the flows on these arcs are bounded above by the customers' inventory capacities (infinite for the supplier). At each period t the supplier receives r^t units of the product and customer i has a demand equal to d_i^t.

The vehicle is represented by one vertex at each period, receiving a zero cost arc from the supplier at the same period with a flow up to Q units, and is then connected

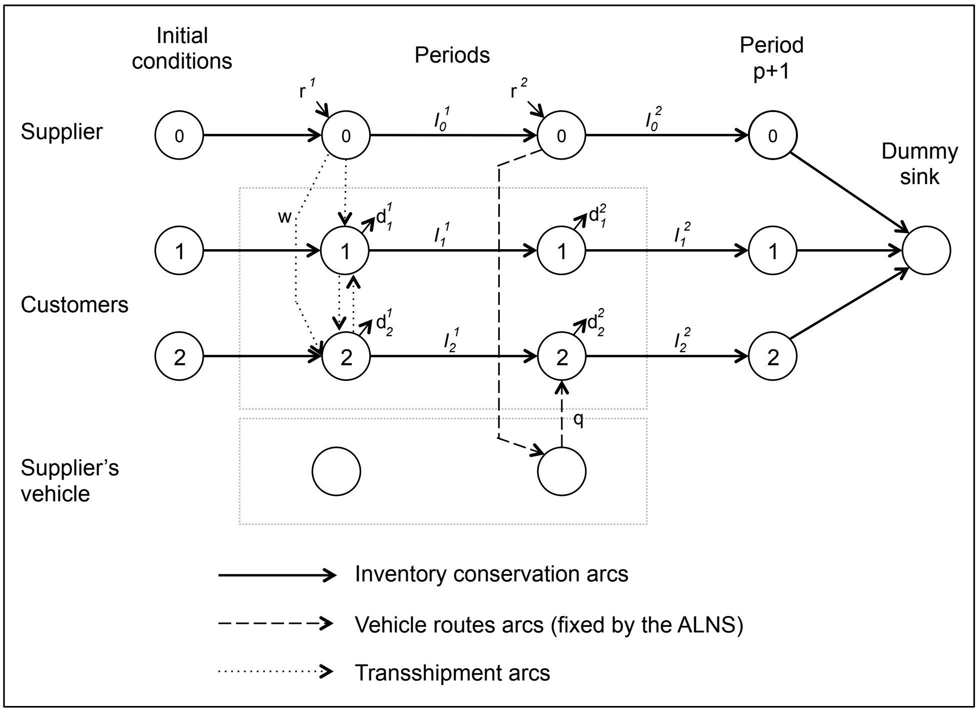

Figure 3.1: Network flow problem for two customers and two time periods.

to each customer receiving a delivery at that period. This is a routing decision made prior to applying the network flow algorithm. They are dashed at period 2, assuming that the routing heuristic has decided to only visit customer 2. We have added dotted arcs representing transshipment options from the supplier and from every customer to every other customer. For the sake of clarity, Figure 1 only shows these arcs for period 1, but these are actually present in all periods. This allows the network flow solution to serve a given customer through a transshipment if a later routing delivery would violate vehicle capacity, or if any inventory constraint is not satisfied by the routing decisions.

The OU policy is enforced by fixing the flow on the arcs linking customers in different successive time periods: once customer i is visited in period t, the arc connecting it to itself at the next period has a flow equal to $C_i - d_i^t$. The network flow algorithm only computes the quantities delivered from all transshipment arcs since the quantities delivered by the supplier are fixed by the OU policy. When the ML replenishment policy is in place, no extra action is needed.

3.3.4 Undirected model for the IRPT

Assuming that the transportation cost matrix is symmetric, we consider an undirected formulation in order to reduce the number of variables. Thus, the model works with the undirected routing variables x_{ij}^t which represent the number of times edge (i, j) is used on the route of the supplier's vehicle in period t. We also introduce variables y_i^t equal to one if and only if node i (the supplier or a customer) is visited at time t, and zero otherwise. The variables w_{ij}^t, I_i^t and q_i^t are the same as in the previous formulations. Then the problem can be formulated as

$$\min \sum_{i \in \mathcal{V}} \sum_{t \in \mathcal{T}} h_i I_i^t + \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}, j<i} \sum_{t \in \mathcal{T}} c_{ij} x_{ij}^t + \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}'} \sum_{t \in \mathcal{T}} b_{ij} w_{ij}^t, \tag{3.18}$$

subject to (3.2)−(3.7) and to:

1. Quantities delivered

$$q_i^t \geq C_i y_i^t - I_i^{t-1} \quad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{3.19}$$

$$q_i^t \leq C_i - I_i^{t-1} \quad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{3.20}$$

$$q_i^t \leq C_i y_i^t \quad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{3.21}$$

2. Routing constraints

$$\sum_{i \in \mathcal{V}'} q_i^t \leq Q y_0^t \quad t \in \mathcal{T}. \tag{3.22}$$

$$\sum_{j \in \mathcal{V}, j<i} x_{ij}^t + \sum_{j \in \mathcal{V}, j>i} x_{ji}^t = 2 y_i^t \quad i \in \mathcal{V} \quad t \in \mathcal{T}. \tag{3.23}$$

$$\sum_{i \in \mathscr{S}} \sum_{j \in \mathscr{S}, j<i} x_{ij}^t \leq \sum_{i \in \mathscr{S}} y_i^t - y_k^t \quad \mathscr{S} \subseteq \mathcal{V}' \quad t \in \mathcal{T}; \tag{3.24}$$

for some $k \in \mathscr{S}$.

3. Integrality and nonnegativity constraints

$$q_i^t, w_{ji}^t \geq 0 \quad i \in \mathcal{V}' \quad j \in \mathcal{V} \quad t \in \mathcal{T}; \tag{3.25}$$

$$x_{i0}^t \in \{0, 1, 2\} \quad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{3.26}$$

$$x_{ij}^t \in \{0, 1\} \quad i, j \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{3.27}$$

$$y_i^t \in \{0, 1\} \quad i \in \mathcal{V} \quad t \in \mathcal{T}. \tag{3.28}$$

As in the previous section, the ML case is modeled by relaxing constraints (3.19).

Archetti et al. (2007) have introduced several classes of inequalities for the IRP. Some of these, namely (17), (18), (19), (21) do not hold in the presence of transshipment. Others, like (22), (23), (24) are still valid for the IRPT, both for the OU and the ML cases. We list them here:

$$x_{i0}^t \leq 2y_i^t \quad i \in \mathcal{V} \quad t \in \mathcal{T}; \tag{3.29}$$

$$x_{ij}^t \leq y_i^t \quad i, j \in \mathcal{V} \quad t \in \mathcal{T}; \tag{3.30}$$

$$y_i^t \leq y_0^t \quad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{3.31}$$

Constraints (3.29) and (3.30) are referred to as logical inequalities. They enforce the condition that if the supplier is the successor of a customer in a route in period t, i.e. $x_{i0}^t = 1$ or 2, then i must be visited, i.e. $y_i^t = 1$. A similar reasoning is applied to customer j in inequalities (3.30). Constraints (3.31) include the supplier in a vehicle route if any customer is visited in that period.

Constraints (20) of Archetti et al. (2007) are modified as follows for the IRPT:

$$\sum_{l=1}^{t} y_i^t \geq \frac{\sum\limits_{l=1}^{t-1} d_i^t - I_i^0 - \sum\limits_{j \in \mathcal{V}} \sum\limits_{l=1}^{t} w_{ji}^t + \sum\limits_{j \in \mathcal{V}'} \sum\limits_{l=1}^{t} w_{ij}^t}{C_i} \quad i \in \mathcal{V} \quad t \in \mathcal{T}. \tag{3.32}$$

Through constraints (3.32) one ensures that customer i has to be visited at least the number of times correspondent to the right-hand side of the inequality. Note that the right-hand side could be rounded up, but this would make the formulation non-linear.

3.3.5 Model with fixed and variable transshipment costs for the IRPT

A meaningful variant of this problem is the one considering a fixed cost for the use of transshipments. This can be achieved through a small modification of the previous model by introducing a new parameter γ equal to the fixed cost of performing a transshipment, and a binary variable z_{ij}^t equal to one if and only if a transshipment takes place in period t from location i to j. Then, the following term must be added to the objective function (3.18):

$$\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}'} \sum_{t \in \mathcal{T}} \gamma z_{ij}^t, \tag{3.33}$$

and the following constraints are added to the model:

$$w_{ij}^t \leq z_{ij}^t C_j \quad i \in \mathcal{V} \quad j \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{3.34}$$

$$z_{ij}^t \in \{0, 1\} \quad i \in \mathcal{V} \quad j \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{3.35}$$

Constraints (3.34) ensure that a transsshipment can only take place if its associated binary variable is set to one, while constraints (3.35) impose its integer condition.

3.4 Branch-and-cut algorithm

The IRPT is $\mathcal{NP}$-hard since it contains the VRP as a special case. If the problem size is relatively small, the undirected formulation can be solved exactly by branch-and-cut as follows. At a generic node of the search tree, a linear program defined by (3.18), (3.2)−(3.7) and (3.19)−(3.23) is solved, a search for violated subtour elimination constraints (3.24) is made and some of these constraints are generated and introduced into the current program which is then reoptimized. This process is repeated until a feasible or dominated solution is reached, or until there are no more cuts to be added and then branching on a fractional variable occurs.

3.4.1 Solution improvement algorithm

The purpose of the Solution Improvement algorithm (SI), is to approximate the cost of a new solution resulting from vertex removals and reinsertions. It is solved whenever the branch-and-cut search identifies a new best solution. Using an idea proposed by Archetti et al. (2012), we simplify and approximate the routing costs resulting from vertex removals and reinsertions as follows. Let a_i^t represent the routing cost reduction if customer i is removed from the route at period t, which obviously visits customer i; let b_i^t represent the routing cost if customer i is inserted in the route at period t, which obviously does not already visit customer i; finally, let r_i^t be a binary parameter equal to 1 if and only if customer i is visited in the current route at period t. Also define the following binary variables: let u_i^t be equal to 1 if and only if customer i is removed from the existing route at period t, and let v_i^t be equal to 1 if and only if customer i is inserted in the route at period t. This subproblem is then to

$$\text{(SI)}\quad \text{minimize} \sum_{t\in\mathcal{T}} h_0 I_0^t + \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} h_i I_i^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} a_i^t u_i^t + \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} b_i^t v_i^t \tag{3.36}$$

subject to (3.2)−(3.7) and to:

$$q_i^t \le C_i - I_i^{t-1} \quad i \in \mathcal{V} \quad t \in \mathcal{T} \tag{3.37}$$

$$q_i^t \le (r_i^t - u_i^t + v_i^t)C_i \quad i \in \mathcal{V}' \quad t \in \mathcal{T} \tag{3.38}$$

$$v_i^t \leq 1 - r_i^t \quad i \in \mathcal{V}' \quad t \in \mathcal{T} \tag{3.39}$$

$$u_i^t \leq r_i^t \quad i \in \mathcal{V}' \quad t \in \mathcal{T} \tag{3.40}$$

$$\sum_{i \in \mathcal{V}'} u_i^t + \sum_{i \in \mathcal{V}'} v_i^t \leq \beta \quad t \in \mathcal{T} \tag{3.41}$$

$$\sum_{i \in \mathcal{V}'} q_i^t \leq Q \quad t \in \mathcal{T} \tag{3.42}$$

$$q_i^t \geq 0 \quad i \in \mathcal{V}' \quad t \in \mathcal{T} \tag{3.43}$$

$$u_i^t, v_i^t \in \{0, 1\} \quad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{3.44}$$

The objective function (3.36) minimizes the total inventory, removal and insertion cost. Constraints (3.37)−(3.38) enforce the ML policy. Constraints (3.39) ensure that if a customer is already present in a route, it cannot be reinserted in the same route. Likewise, constraints (3.40) guarantee that only those customers belonging to a route can be removed from it. Constraints (3.42) ensure that the vehicle capacity is respected. If the incumbent solution is changed by more than one customer, then this model only provides an approximation of the actual routing costs. For this reason, we have decided to limit the number of insertions and removals that could take place in the solution of SI, and we have added constraints (3.41) to limit the number of insertions and removals per route to a small value β.

We provide a simplified formal description of the method in Algorithm 3.1.

3.4.2 Implementation details

We offer a few remarks and comments regarding the implementation of the algorithm. We have implemented both the directed and the undirected formulations, but none outperforms the other significantly. We have opted for the edge formulation because it requires considerably fewer variables and this becomes a relevant issue on large instances.

In order to solve the LP relaxation at each node we use the dual simplex algorithm. In our tests it has shown to outperform the primal simplex method.

A major difference between our implementations and that of Archetti et al. (2007) is that we do not compute an upper bound at the beginning of the search. Archetti et al. (2007) uses the heuristic of Bertazzi et al. (2002), which is known to produce reasonably good starting solutions in very short time. We, in contrast, apply an algorithm to further improve integer solution found during the search, thus helping find better solutions faster. This algorithm, described in Section 3.4.1 is an approximation of the true routing costs.

Algorithm 3.1 Proposed branch-and-cut algorithm

```
At the root node of the search tree, generate and insert all valid inequalities
(3.29)−(3.32) into the program.
Subproblem solution. Solve the LP relaxation of the node.
Termination check:
if there are no more nodes to evaluate then
  Stop.
else
  if The current solution is a new best solution then
    Apply the SI algorithm to the incumbent solution.
    if the SI algorithm yields an improved solution then
      Update the solution vector at the branch-and-cut level
    end if
  end if
  Select one node from the branch-and-bound tree.
end if
while the solution of the current LP relaxation contains subtours do
  Identify the connected components using the separation procedure of Padberg
  and Rinaldi (1991).
  Add all violated subtour elimination constraints (3.24).
  Subproblem solution. Solve the LP relaxation of the node.
end while
if the solution of the current LP relaxation is integer then
  Go to the termination check.
else
  Branching: branch on one of the fractional variables.
  Go to the termination check.
end if
```

3.5 Adaptive large neighborhood search heuristic

A heuristic is needed for instances of realistic size. Because the problem combines several dimensions (routing, scheduling, inventory management and transshipment), a powerful metaheuristic is required for its solution. Such a metaheuristic is the ALNS framework recently proposed by Ropke and Pisinger (2006a) for the VRP and applied to a number of other contexts (Bartodziej et al., 2009; Hewitt et al., 2010; Laporte et al., 2010; Pepin et al., 2009). This type of algorithm is highly suitable for the problem at hand because of its generality and flexibility. It can simultaneously handle several families of hard constraints and it conducts a highly diversified search through the multiplicity of its operators and through the use of a random mechanism for their selection.

We now describe our ALNS heuristic. The algorithmic framework is made up of five main components.

1. **Large neighborhood:** At each iteration, a number of customers are removed from their current route and are eventually reinserted. This fixes the decisions regarding routing, and the problem is passed to a network flow solver to optimize all remaining decisions simultaneously (minimize total costs taking into account inventory holding costs, transshipments and delivery quantities), as described in Section 3.3.3.

2. **Adaptive search engine:** The choice of which operator to apply at a given iteration is governed by a roulette-wheel mechanism in which each operator is assigned a weight depending on its past performance. Let ω_i be a measure of how well operator i has performed in the past; then given h operators with weights ω_i, operator j will be selected with probability $\omega_j / \sum\limits_{i=1}^{h} \omega_i$.

3. **Adaptive weight adjustment:** The search is divided into segments of φ iterations each, and weights are computed by taking into account the performance of the operators during the last segment. Each operator is assigned a weight and a score. Initially, all weights are equal to one and all scores are equal to zero. At each iteration, scores are updated as follows: if an operator finds a new best solution, its score is increased by σ_1; if it finds a solution better than the incumbent, its score is increased by σ_2; if the solution is not better but is still accepted, the score is increased by σ_3. Obviously $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$. After φ iterations, the weights are updated considering the scores obtained in the last segment and the scores are reset to zero. To do so, let π_i and o_{ij} be, respectively, the score of the operator i and the number of times operator i

has been used in the last segment j, normalized by a factor $\nu_i \geq 1$ reflecting the computational effort it requires (see Ropke and Pisinger (2006b)). The *normalization factor* ν_i multiplies o_{ij}, and therefore decreases the weight of operator i, so that the more time consuming operators are applied less frequently. The values used for the normalization factors are all equal to one in our implementation, except for three cases where different values are used. These are provided in Sections 4.2.3, 4.2.4 and 4.2.12. The updated weights are then

$$\omega_i := \begin{cases} \omega_i & \text{if } o_{ij} = 0 \\ (1-\eta)\omega_i + \eta\pi_i/\nu_i o_{ij} & \text{if } o_{ij} \neq 0, \end{cases} \tag{3.45}$$

where $\eta \in [0, 1]$ is called the reaction factor, controlling how quickly the weight adjustment reacts to changes in the operator performance.

4. **Periodic postoptimization:** At the end of each segment, we apply a 2-opt procedure to each vehicle route.

5. **Acceptance and stopping criteria:** As in Ropke and Pisinger (2006a), we use an acceptance criterion based on simulated annealing. Given a solution s, a neighbor solution s' is accepted if $z(s') < z(s)$, and with probability $e^{-(z(s')-z(s))/\tau}$ otherwise, where $z(s)$ is the solution cost defined by (3.1) and $\tau > 0$ is the current temperature. The temperature starts at τ_{start} and is decreased by a cooling rate factor ϕ at each iteration, where $0 < \phi < 1$. To avoid long computations for large and difficult instances, we limit the running time to one hour and also to a maximum number of iterations, as described in Section 3.5.3. The use of simulated annealing not only prevents the search mechanism from cycling, but it also provides an added diversification effect.

3.5.1 Initial solution

The initial solution is generated by randomly selecting 75% of the customers and randomly assigning each of them to a period of the planning horizon. Their insertion in the routes follows the cheapest insertion rule. Our computational experiments have shown that the initial solution does not have a significant impact on the overall solution cost or on the running time.

3.5.2 List of operators

Unlike related problems such as the VRP, where every removal is accompanied by an insertion, one may decide to remove a vertex from some periods and not reinsert it back. This partial solution will still be feasible when transshipments are allowed. Moreover, in the IRPT, it is feasible and sometimes optimal not to create any route because transshipments can always be made. This observation has motivated us not to use the traditional destroy and repair framework of Shaw (Shaw, 1997) and Pisinger and Ropke (Pisinger and Ropke, 2007) in which each destroy operator is always followed by a repair operator, but to keep the option of making only a removal or only an insertion. In the operators described in Sections 3.4.2.3, 3.4.2.4 and 3.4.2.12, it is implicitly assumed that solution costs are computed through the network flow algorithm of Section 3.3.3 once a route is fixed. In Sections 3.4.2.2, 3.4.2.6 3.4.2.8 and 3.4.2.11, the best insertion position is found by computing the insertion cost at each position in the route, which has linear complexity with respect to the number of customers in the route. In what follows, ρ is an integer randomly drawn from the interval $[1,\ n]$ using a semi-triangular distribution with a negative slope.

3.5.2.1 Randomly remove ρ

This operator randomly selects one period and randomly removes one customer from it. The complexity of this operator is $O(1)$. It is repeated ρ times. This operator is useful for refining the solution since it does not change it much when ρ is small (which happens frequently). However, it still yields a major transformation of the solution when ρ is large.

3.5.2.2 Randomly insert ρ

This operator randomly selects unrouted customers, up to a maximum of ρ, and one random period for each of them. It inserts the customer in the best position in the route of the selected period. The complexity of each insertion is $O(n)$.

3.5.2.3 Remove worst ρ

This operator removes the customer that will save the most when removed, considering the total routing, inventory and transshipment cost. It is repeated ρ times. Because a network flow problem must be solved for each customer, the complexity of each removal is $O(nf(n))$, where $f(n)$ is the complexity of solving the min-cost network flow problem. The normalization factor used for this operator is 50.

3.5.2.4 Insert best ρ

This operator is analogous to the previous one. It is repeated ρ times by computing the cheapest insertion with respect to total costs.

3.5.2.5 Shaw removal

Following the ideas developed by Ropke and Pisinger (2006a) and by Shaw (1997), this operator removes customers that are relatively close to each other. Specifically, this heuristic randomly selects one period from the planning horizon and one customer served in this period, computes the distance $dist_{min}$ to the closest customer also being served by the same route, and removes all customers within $2dist_{min}$ units from the selected route. The complexity of this operator is $O(n)$.

3.5.2.6 Shaw insertions

This operator is similar to the Shaw removal in the sense that it selects similar customers to be inserted together. It selects one period and one customer not served in that period. The heuristic then computes $dist_{min}$ and all customers within a $2dist_{min}$ distance are inserted in the same period, always following the cheapest insertion rule. The complexity of this operator is $O(n)$.

3.5.2.7 Remove ρ customers

This operator removes a customer from all routes where it appears. Its complexity is $O(p)$ since each customer appears in at most p routes. It is repeated ρ times. The motivation for this operator is to allow these customers to be assigned to different sets of periods.

3.5.2.8 Insert ρ customers

This operator iteratively selects ρ customers and assigns them to the best position of the vehicle route in several randomly selected periods if the customer is not yet present in these periods. Its complexity is $O(np)$ since each selected customer will be inserted in at most p periods. The motivation is to diversify the search towards unexplored areas of the search space by allowing customers to be served in different periods from the ones currently selected.

3.5.2.9 Empty one period

This operator randomly selects one period and removes all customers from it. It is implemented in $O(1)$ time. The motivation is to allow different periods to have opened routes.

3.5.2.10 Swap routes

This operator randomly selects two periods and swaps their routes. It is implemented in $O(1)$ time.

3.5.2.11 Randomly move ρ

This operator selects one period and one customer being served in this period, removes it and serves it in the best possible position in a different randomly selected period. Its complexity is $O(n)$. It is repeated ρ times.

3.5.2.12 Multiple interchanges

This operator is used only in IRP-OU and IRPT-OU since we observed that for these variants the remaining operators were not sufficient: due to the OU policy, the insertion of a customer in the incumbent solution has a great impact on the inventory costs and on vehicle loads. This operator performs interchanges like the heuristic of Bertazzi et al. (2002) summarized in Chapter 2, but differs from the original algorithm in three ways: 1) in order to limit the computational burden, it does not iterate as long as improvements can be obtained, but stops after $n/2$ iterations; 2) it does not restart from an empty solution at each call, but is initiated from the current solution; 3) it applies some improvement procedures to take transshipments into account as they were not present in the original algorithm: if a customer has a lower inventory cost than that of the supplier, products are transshipped to that customer if the trade-off is positive, while respecting its inventory holding capacity; transshipments are combined to an early period if this yields savings, and routes from the last two periods are replaced by transshipments, since one could save on inventory holding costs if the OU policy did not have to be enforced. Delivery quantities are determined as in Bertazzi et al. (2002) by computing shortest paths on acyclic networks $\mathcal{N}_i$, one for each customer i. Each node of $\mathcal{N}_i$ corresponds to a discrete time instant between 0 and $p + 1$, and arc (t, t') is defined if no stockout occurs at customer i whenever it is not visited in the interval $[t, t']$; the quantity delivered to i at each time period will be that to fill the customer capacity and the arcs cost is the sum of the inventory and routing costs associated with visiting customer i in the interval $[t, t']$. Algorithm 3.2 provides a formal description of this operator. This operator can be implemented in $O(n^2p^2)$ time if Dijkstra's algorithm (Dijkstra, 1959) is used to compute the shortest paths. The normalization factor used for this operator is 20.

Algorithm 3.2 Multiple interchanges operator

```
1: Sort the set of customers $\mathcal{V}'$ in the non-decreasing order of the ratio between $C_i$
   and $\sum_{t \in \mathcal{T}} d_i^t/p$. Relabel them accordingly.
2: Let $z(s)$ be the cost of the incumbent solution $s$.
3: $iterations \leftarrow 0$.
4: while $iterations < n/2$ and improvements are made do
5:   for $i = 1, \ldots, n$ do
6:     for $j = n, \ldots, 1$ and $j \neq i$ do
7:       $s' \leftarrow s$.
8:       Remove customers $i$ and $j$ from $s'$.
9:       Create an acyclic network $\mathcal{N}_j$ for customer $j$.
10:      Solve the shortest path over $\mathcal{N}_j$ from 0 to $p+1$ and insert customer $j$ in
         the periods represented by the selected nodes.
11:      Transship goods to $j$ up to $U_j$ if the trade-off (transshipping cost − inventory cost) is positive.
12:      Combine transshipments to $j$ to an early period if this yields savings.
13:      Replace deliveries from the last two periods by transshipments.
14:      Repeat steps 9 to 13 for customer $i$.
15:      if $z(s') < z(s)$ then
16:        $s \leftarrow s'$;
17:      end if
18:    end for
19:  end for
20: end while
21: return $s$.
```

3.5.3 Parameter settings and pseudocode

We now describe our ALNS pseudocode and the parameters that govern the algorithm. We have tested different combinations for the parameters during a tuning phase, mostly through an ad hoc trial and error phase in the development of the heuristic. The starting temperature τ_{start} is set to 30,000 and the cooling rate ϕ is 0.9994, which yields roughly 25,000 iterations, our desired number of repetitions. The stopping criterion is satisfied when the temperature reaches 0.01, when 25,000 iterations have been performed, or when 3,600 seconds have elapsed. Adjusting the cooling mechanism based on the number of iterations alone may not work for large instances which require more computing time per iteration. In such cases, the temperature may be too high when the time limit is reached. This is why we decrease it not only on the basis of the iteration count, but also on the basis of the elapsed computation time. In our implementation, the segment length φ was set to 200 iterations and the reaction factor η was set to 0.7, that is, new weights will be composed by 70% of the performance on the last segment and 30% by the last weight value. Scores are updated with $\sigma_1 = 10$, $\sigma_2 = 5$ and $\sigma_3 = 2$. At the end of each segment we also perform the periodic postoptimization described in item 4 at the beginning of Section 3.5.

Algorithm 3.3 shows the pseudocode for our ALNS.

3.5.4 ALNS applied to the IRPT-OU

In this section we briefly describe how our ALNS is implemented for the IRPT-OU. Once the ALNS has fixed the routing variables, the remaining problem is modeled as a network flow problem and solved by means of a specialized minimum cost network flow algorithm, as described in Section 3.3. It is easy to see that if all transshipment variables are set to zero, the problem reduces to IRP-OU which yields an upper bound on the IRPT-OU optimum.

In Figure 3.1, this corresponds to fixing the flow on the horizontal inventory conservation arcs to meet the OU policy, i.e. $I_i^t = C_i - d_i^t$ if customer i is set to be visited by the vehicle.

3.5.5 ALNS applied to the IRPT-ML

This version of the problem is similar to the IRPT-OU except that the arcs connecting customers between successive time periods do not force the flow to respect the OU policy. The network flow algorithm determines the quantities delivered by the vehicle and from all transshipment arcs. Once again, if all transshipment variables

Algorithm 3.3 ALNS heuristic

```
Initialize: set all weights equal to 1 and all scores equal to 0.
$s_{best} \leftarrow s \leftarrow$ *initial solution.*
$\tau \leftarrow \tau_{start}$.
while $\tau > 0.01$ and $time < 3{,}600$ and $iterations < 25{,}000$ do
    $s' \leftarrow s$.
    Select an operator using the roulette-wheel mechanism based on the current weights.
    Apply the operator to $s'$ and update the number of times it is used.
    if $z(s') < z(s)$ then
        $s \leftarrow s'$;
        if $z(s) < z(s_{best})$ then
            $s_{best} \leftarrow s$;
            update the score for the operator used with $\sigma_1$;
        else
            update the score for the operator used with $\sigma_2$;
        end if
    else if $s'$ is accepted by the simulated annealing criterion then
        $s \leftarrow s'$;
        update the score for the heuristic used with $\sigma_3$.
    end if
    if the iteration count is a multiple of $\varphi$ then
        update the weights of all operators and reset their scores.
        perform an intra-route 2-opt to improve the sequence of customers.
    end if
    if $time > 1{,}200$ and $iterations < 25{,}000/3$ then
        $\phi \leftarrow (0.01/\tau)^{1/(2\cdot iterations)}$;
    end if
    if $time > 2{,}700$ and $iterations < 25{,}000/2$ then
        $\phi \leftarrow (0.01/\tau)^{1/(iterations/2)}$;
    end if
    $\tau \leftarrow \phi\tau$;
end while
return $s_{best}$;
```

are zero, the problem reduces to an IRP-ML which yields an upper bound on the IRPT-ML optimum. This variant entails no changes in the network depicted in Figure 3.1.

3.5.6 ALNS applied to the IRP-OU

We have applied our algorithm to the IRP-OU without any structural change. To avoid passing an infeasible problem to the network flow algorithm (for instance when vehicle capacity would be exceeded or when a stockout would occur at a customer due to it not being served as often as required), we have kept all transshipment arcs from the supplier and from every customer to every other customer, with large artificial costs; this means that feasible solutions can always be reached, but at very high cost if transshipments are used. These costs act as penalties in the objective function when the vehicle capacity is exceeded or when the master level heuristic does not add all customers to the current solution.

The remaining problem is then similar to the IRPT-OU: decisions regarding routings are fixed by the ALNS algorithm and modeled as a network flow problem with one vertex representing the vehicle for each period, and arcs leaving the vehicle vertex and arriving at each selected customer. The vehicle vertex receives an arc from the supplier with up to Q units of flow. The OU policy is modeled on the network of Figure 3.1 by fixing the flow on the horizontal arcs connecting customers in successive time periods: once customer i is visited in period t, the arc linking to it in the next period has a flow equal to $C_i - d_i^t$. In order to prevent transshipments, but still allow feasible solutions to be reached irrespective of the routing decisions, all transshipment arcs are assigned a high unit cost.

3.5.7 ALNS applied to the IRP-ML

Modeling the IRP-ML as a network flow problem is similar to the IRP-OU, except that arcs connecting the customers in successive time periods have a minimum flow equal to 0. The vehicle vertex is fed from the supplier with up to Q units and the minimum-cost network flow algorithm decides on how much to deliver to each of the customers selected from the master level heuristic. Dummy arcs are again inserted to penalize unvisited customers and solutions in which the vehicle capacity would be exceeded. It is easy to see that the IRP-OU yields an upper bound on the IRP-ML optimum as we just relaxed one constraint of the former problem. The only modification on the network of Figure 3.1 concerns the cost of transshipment arcs as in the previous case.

3.6 Computational results

Our ALNS algorithm was coded in C++ using Microsoft Visual Studio 2008. We used the scaling push-relabel algorithm for the minimum-cost flow problem developed by Goldberg (1997) to solve the second level problem. It was run on an Asus F8s Intel T7700 Core2Duo 2.4GHz and 4 GB RAM laptop PC. The branch-and-cut algorithm was coded in C++ using IBM Concert Technology and CPLEX 12.3 with six threads. Its computations were executed on a grid of Intel Xeon™ processors running at 2.66 GHz with up to 48 GB of RAM installed per node, with the Scientific Linux 6.1 operating system.

To evaluate the performance of our algorithms, we have used the instances of the IRP generated and solved to optimality by Archetti et al. (2007). These instances are divided into two classes according to their inventory cost, as in Archetti et al. (2007). In low cost instances, inventory holding costs are selected randomly in the interval [0.01, 0.05]; in high cost instances, inventory holding costs are selected randomly in the interval [0.1, 0.5]. We report average statistics over five instances for each combination. The instance data are identical to those of Archetti et al. (2007), but we now allow transshipments, as described in Section 3.3.

There are no previously reported solutions for the IRPT since we are introducing the problem in this paper. We have compared our ALNS algorithm against the optimal solutions obtained with the model described in Section 3.4. In Tables 3.1 and 3.2 we report the optimal solutions obtained by the branch-and-cut algorithm and the solution values obtained by the ALNS heuristic for the IRPT-OU, as well as the gaps with respect to the best known lower bounds on the optimal solutions, found by the branch-and-cut algorithm. Results for the IRPT-ML are reported in Tables 3.3 and 3.4. For full results on all instances, the reader is referred to Coelho et al. (2011a) and to Appendix A.1 for heuristic solutions and Appendix A.3 for exact solution values.

Out of the 160 instances tested, our branch-and-cut algorithm was able to match the solution values on 61 and improved the solution values on 99 of them. It was able to provide optimal solutions for most of the instances.

We have also applied our algorithms to the traditional IRP (without transshipment) by setting the transshipment cost sufficiently large ($b_{ij} = c_{ij}$) so as to avoid the use of transshipment in the final solution. Our branch-and-cut algorithm was able to quickly obtain optimal solutions for the single-vehicle case. It is difficult if not impossible to make a clear comparison with the solution times obtained exactly by Archetti et al. (2007) and heuristically by Archetti et al. (2012) due to differences

Table 3.1: Average results for the IRPT-OU $-$ $p = 3$ and transshipment cost $b_{ij} = 0.01c_{ij}$

		Branch-and-cut				ALNS		
	Instance	# solved	UB	gap (%)	time (s)	z	gap (%)	time (s)
Low inventory cost	absn05	5	745.39	0.00	0.2	745.39	0.00	6.56
	absn10	5	1616.12	0.00	1.0	1617.51	0.08	29.55
	absn15	5	1851.14	0.00	1.6	1864.70	0.73	82.24
	absn20	5	2328.91	0.00	5.0	2442.44	4.87	183.28
	absn25	5	2608.80	0.00	8.6	2724.67	4.44	389.10
	absn30	5	3002.36	0.00	17.2	3341.49	11.29	635.79
	absn35	5	3232.49	0.00	44.8	3522.66	8.97	895.11
	absn40	5	3350.06	0.00	88.6	3795.60	13.29	1577.21
	absn45	5	3563.46	0.00	253.4	4078.89	14.46	2350.23
	absn50	5	3915.22	0.00	1006.6	4581.07	17.00	2898.06
	Average	5		0.00	142.7		7.51	904.71
High inventory cost	absn05	5	1664.38	0.00	0.4	1664.38	0.00	8.06
	absn10	5	4044.12	0.00	1.0	4044.12	0.00	30.12
	absn15	5	5069.51	0.00	1.4	5116.21	0.92	74.63
	absn20	5	6865.59	0.00	3.8	6927.36	0.89	151.21
	absn25	5	8591.75	0.00	7.2	8754.03	1.88	350.23
	absn30	5	10585.63	0.00	15.6	10867.17	2.65	521.52
	absn35	5	11161.39	0.00	34.6	11367.04	1.84	930.08
	absn40	5	12146.58	0.00	201.2	12563.16	3.42	1577.64
	absn45	5	13527.38	0.00	262.8	13921.34	2.91	2244.01
	absn50	5	14943.66	0.00	752.0	15560.06	4.12	3375.50
	Average	5		0.00	128.0		1.86	926.30

Table 3.2: Average results for the IRPT-OU $-$ $p = 6$ and transshipment cost $b_{ij} = 0.01c_{ij}$

		Branch-and-cut				ALNS		
	Instance	# solved	UB	gap (%)	time (s)	z	gap (%)	time (s)
Low inventory cost	absn05	5	2561.85	0.00	1.0	2561.85	0.00	16.23
	absn10	5	4011.20	0.00	4.2	4095.55	2.10	70.29
	absn15	5	4744.54	0.00	18.0	4881.79	2.89	208.08
	absn20	5	5755.46	0.00	673.0	6276.32	9.05	491.31
	absn25	4	6335.43	0.19	10845.0	6806.59	7.65	805.16
	absn30	1	7092.57	2.99	35330.4	7981.14	16.11	1650.37
	Average	4.16		0.53	7811.9		6.30	540.24
High inventory cost	absn05	5	4759.54	0.00	0.6	4760.94	0.02	14.98
	absn10	5	7990.54	0.00	3.2	8038.53	0.60	65.29
	absn15	5	10858.99	0.00	16.6	11027.72	1.55	156.07
	absn20	5	13735.66	0.00	379.2	14278.32	3.95	442.43
	absn25	4	16000.70	0.20	11170.2	16867.70	5.64	906.47
	absn30	2	19738.40	0.39	31120.6	20492.18	4.26	1718.08
	Average	4.33		0.09	7115.0		2.67	550.55

Table 3.3: Average results for the IRPT-ML $-$ $p = 3$ and transshipment cost $b_{ij} = 0.01c_{ij}$

		Branch-and-cut				ALNS		
	Instance	# solved	UB	gap (%)	time (s)	z	gap (%)	time (s)
Low inventory cost	absn05	5	744.89	0.00	0.2	744.89	0.00	5.28
	absn10	5	1577.31	0.00	0.8	1586.91	0.60	14.52
	absn15	5	1840.07	0.00	1.2	1849.83	0.53	30.64
	absn20	5	2278.04	0.00	4.0	2290.55	0.54	56.74
	absn25	5	2578.75	0.00	8.4	2579.18	0.01	92.04
	absn30	5	2964.08	0.00	12.6	2985.99	0.73	165.20
	absn35	5	3200.62	0.00	29.0	3448.17	7.73	248.74
	absn40	5	3310.14	0.00	58.6	3361.80	1.56	348.93
	absn45	5	3519.90	0.00	107.0	3697.61	5.04	461.71
	absn50	5	3861.28	0.00	520.0	4071.09	5.43	760.30
	Average	5		0.00	74.1		2.22	218.41
High inventory cost	absn05	5	1660.27	0.00	0.2	1660.27	0.00	5.98
	absn10	5	3999.03	0.00	0.8	4011.81	0.31	15.36
	absn15	5	5054.50	0.00	1.2	5061.92	0.14	33.76
	absn20	5	6818.76	0.00	3.4	6869.26	0.74	57.73
	absn25	5	8557.89	0.00	6.4	8562.59	0.05	91.26
	absn30	5	10533.45	0.00	11.4	10557.63	0.22	159.13
	absn35	5	11121.67	0.00	23.8	11309.46	1.68	282.47
	absn40	5	12095.24	0.00	57.4	12165.28	0.57	360.83
	absn45	5	13458.36	0.00	121.6	13699.96	1.79	627.96
	absn50	5	14892.42	0.00	579.2	15004.18	0.75	939.87
	Average	5		0.00	80.5		0.63	257.43

Table 3.4: Average results for the IRPT-ML $-$ $p = 6$ and transshipment cost $b_{ij} = 0.01c_{ij}$

		Branch-and-cut				ALNS		
	Instance	# solved	UB	gap (%)	time (s)	z	gap (%)	time (s)
Low inventory cost	absn05	5	2554.45	0.00	0.6	2558.37	0.15	10.90
	absn10	5	3978.71	0.00	3.8	4095.10	2.92	36.70
	absn15	5	4724.20	0.00	14.0	4834.73	2.34	80.68
	absn20	5	5715.93	0.00	348.0	6020.83	5.33	174.24
	absn25	5	6294.01	0.00	6183.4	6808.40	8.17	295.30
	absn30	4	7034.53	0.57	31961.0	7466.89	6.76	671.24
	Average	4.83		0.09	6418.4		4.28	211.51
High inventory cost	absn05	5	4742.19	0.00	0.6	4748.31	0.12	13.28
	absn10	5	7940.05	0.00	2.8	7961.72	0.27	37.91
	absn15	5	10819.69	0.00	9.8	10949.14	1.19	88.45
	absn20	5	13678.28	0.00	279.8	14152.04	3.46	179.70
	absn25	5	15937.70	0.00	11677.6	16320.18	2.40	329.51
	absn30	4	19661.20	0.07	34255.6	20235.5	3.00	787.47
	Average	4.83		0.01	7704.3		1.74	239.38

in CPLEX versions and on the hardware used. However, on instances with 30 customers and six time periods, our algorithm took on average 70 seconds compared to 1570 seconds of Archetti et al. (2007) and 1922 seconds of Archetti et al. (2012). Our heuristic algorithm was also able to solve most of the short period instances with several vehicles. We have used the same time limit of 3,600s. In Tables 3.5 and 3.6, our results are compared to those of Archetti et al. (2012) and of Bertazzi et al. (2002) on these IRP-OU instances. Our results are significantly better than those of Bertazzi et al. (2002) but slightly worse than those of Archetti et al. (2012). The instances reported in the BPS (Bertazzi et al., 2002) column were in fact generated and solved by Archetti et al. (2012) using the code of Bertazzi et al. (2002). The CPU times for the solution of these instances are not provided but are reported to be very small (a few seconds for $n \leq 100$ and less than three minutes for $n \leq 200$). We have also tested our algorithm with the IRP-ML and the gaps to the optimal solutions were slightly larger than those observed in the IRP-OU. The machines used in Archetti et al. (2012) and in our experiments are different; according to SPEC (SPEC) our computer is approximately 30% faster than the one used by Archetti et al. (2012).

We have evaluated the impact of the transshipment cost on the solution and its cost. To this end, we have gradually increased the transshipment cost, and we have solved a subset of instances under the IRPT-OU and IRPT-ML policies using the ALNS heuristic. The results of these experiments are reported in Tables 3.7 and 3.8. It can be seen that solution values become much closer to those of the traditional IRP when the transshipment cost increases from $b_{ij} = 0.01c_{ij}$ to $0.05c_{ij}$, and several instances make no use of transshipment when b_{ij} is set equal to $0.10c_{ij}$. Transshipments start to be economically interesting when the cost of outsourcing the delivery of ten units does not exceed the cost of transporting one unit with the supplier's vehicle, all other costs being identical. It should be noted, however, that this conclusion may not extend to real-life instances which are often different from artificial ones.

Moreover, we have performed tests to assess the impact of individual operators in our ALNS heuristic. We have examined the impact of removing individual operators for a subset of 32 instances, including small, medium and large ones, with three and six time periods. Specifically, we have selected the first instance of each size: $n = 5, 10, 15, 20, 25, 30, 35, 40, 45, 50$ for $p = 3$, and $n = 5, 10, 15, 20, 25, 30$ for $p = 6$, both for low and high inventory costs. The results of the experiments for the IRPT-ML are summarized in Table 3.9.

Removing some operators can have a major impact both on solution quality and

Table 3.5: Average heuristic results for the IRP-OU $- p = 3$

		ABLS	BPS		ABHS			ALNS		
	Instance	$z*$	z	gap (%)	z	gap (%)	time$_1$ (s)	z	gap (%)	time$_2$ (s)
Low inventory cost	absn05	1418.75	1465.75	2.88	**1418.75**	**0.00**	3	**1418.75**	**0.00**	10.71
	absn10	2228.66	2245.61	0.78	2228.72	0.00	12.8	**2228.66**	**0.00**	35.28
	absn15	2493.47	2555.21	2.56	**2493.47**	**0.00**	41.4	**2493.47**	**0.00**	99.32
	absn20	3053.01	3176.91	3.83	**3053.55**	**0.02**	104.2	3055.58	0.09	239.76
	absn25	3451.14	3552.08	2.99	**3451.14**	**0.00**	258.8	3451.86	0.02	572.28
	absn30	3643.21	3774.20	3.60	**3643.99**	**0.02**	515.00	3645.70	0.07	1072.47
	absn35	3846.86	4022.04	4.46	**3848.46**	**0.04**	808.80	3850.83	0.10	1439.28
	absn40	4125.70	4394.94	6.46	**4128.50**	**0.07**	1168.60	4140.16	0.35	2755.72
	absn45	4270.61	4594.91	7.60	**4276.89**	**0.14**	1460.00	4283.33	0.30	3417.87
	absn50	4810.86	5090.68	5.81	**4831.97**	**0.44**	2280.60	4841.26	0.64	2675.47
	Average			4.09		**0.07**	665.32		0.15	1231.81
High inventory cost	absn05	2354.17	2393.09	1.31	**2354.17**	**0.00**	4.60	**2354.17**	**0.00**	9.45
	absn10	4690.46	4774.67	1.74	**4691.02**	**0.01**	13.20	**4691.02**	**0.01**	34.62
	absn15	5736.90	5858.66	2.18	**5738.11**	**0.02**	46.60	5740.66	0.07	97.09
	absn20	7619.91	7870.47	3.30	**7620.59**	**0.01**	104.20	7626.94	0.09	224.24
	absn25	9460.74	9554.62	1.06	**9460.74**	**0.00**	222.00	9476.04	0.17	446.47
	absn30	11320.63	11460.87	1.21	**11342.08**	**0.17**	431.60	11354.66	0.28	890.16
	absn35	11828.80	12096.13	2.25	**11842.24**	**0.09**	833.40	11848.90	0.19	1600.56
	absn40	13011.45	13315.08	2.26	**13011.45**	**0.00**	1293.60	13043.95	0.26	2767.76
	absn45	14317.82	14669.39	2.49	**14322.96**	**0.04**	1534.40	14392.04	0.52	3010.08
	absn50	15948.78	16198.76	1.57	**15975.00**	**0.17**	2830.20	16077.86	0.81	2987.26
	Average			1.93		**0.05**	731.38		0.24	1206.76

time$_1$: run on an Intel Dual Core 1.86GHz and 3.2 GB RAM

time$_2$: run on an Intel Core2Duo 2.4GHz and 4 GB RAM

Table 3.6: Average heuristic results for the IRP-OU $-$ $p = 6$

		ABLS	BPS		ABHS			ALNS		
	Instance	z*	z	gap (%)	z	gap (%)	time$_1$ (s)	z	gap (%)	time$_2$ (s)
Low inventory cost	absn05	3299.97	3348.43	1.64	**3299.97**	**0.00**	17.80	**3299.97**	**0.00**	20.92
	absn10	4832.87	4899.85	1.36	**4832.87**	**0.00**	76.80	**4832.87**	**0.00**	95.88
	absn15	5566.37	5803.08	4.27	**5566.37**	**0.00**	337.40	5582.80	0.28	337.70
	absn20	6833.27	7035.02	2.95	**6838.41**	**0.08**	837.80	6857.90	0.39	797.63
	absn25	7454.14	7913.47	6.19	**7471.41**	**0.23**	1720.00	7487.80	0.45	1610.54
	absn30	7847.37	8214.21	4.64	7892.28	0.56	3321.00	**7888.56**	**0.53**	3031.66
	Average			3.50		**0.14**	1051.80		0.27	982.38
High inventory cost	absn05	5538.01	5555.91	0.34	**5538.01**	**0.00**	19.80	5538.91	0.02	22.82
	absn10	8872.41	9036.86	1.87	**8872.41**	**0.00**	90.40	**8872.41**	**0.00**	106.84
	absn15	11721.83	11852.95	1.20	**11721.83**	**0.00**	289.00	11738.50	0.14	370.67
	absn20	14863.85	15179.46	2.09	**14882.83**	**0.13**	746.60	14883.49	0.13	1021.13
	absn25	17170.80	17534.01	2.12	**17191.87**	**0.12**	1781.40	17223.47	0.31	2221.46
	absn30	20657.29	21180.91	2.55	**20705.65**	**0.25**	3164.60	20752.32	0.48	3399.49
	Average			1.69		**0.08**	1015.30		0.18	857.06

time$_1$: run on an Intel Dual Core 1.86GHz and 3.2 GB RAM

time$_2$: run on an Intel Core2Duo 2.4GHz and 4 GB RAM

Table 3.7: Average increase of the solution cost when the transshipment cost increases from $b_{ij} = 0.01c_{ij}$ to $0.05c_{ij}$ for the IRPT-OU

Set of instances	Avg gap (%) to the IRP-OU	Avg gap (%) to the IRP-OU
	when $b_{ij} = 0.01c_{ij}$	when $b_{ij} = 0.05c_{ij}$
$p = 3$, low inventory cost	-20.38	-2.31
$p = 3$, high inventory cost	-11.89	-1.39
$p = 6$, low inventory cost	-10.25	-0.19
$p = 6$, high inventory cost	-6.06	0.00

Table 3.8: Average increase of the solution cost when the transshipment cost increases from $b_{ij} = 0.01c_{ij}$ to $0.05c_{ij}$ for the IRPT-ML

Set of instances	Avg gap (%) to the IRP-ML	Avg gap (%) to the IRP-ML
	when $b_{ij} = 0.01c_{ij}$	when $b_{ij} = 0.05c_{ij}$
$p = 3$, low cost	-18.21	-0.06
$p = 3$, high cost	-9.63	-0.21
$p = 6$, low cost	-10.75	-0.09
$p = 6$, high cost	-4.96	0.21

Table 3.9: Average increase (%) of the solution cost when individual operators are removed from the ALNS algorithm - IRPT-ML

Operator removed	1	2	3	4	5	6	7	8	9	10	11
Average increase (%)	0.47	1.63	0.52	1.38	0.26	0.39	0.94	0.16	2.46	0.35	0.09

on running time. Specifically, operators that require solving a network flow problem frequently are time consuming. Removing them reduces the CPU time significantly at the expense of a slight decrease in solution quality. Also, operators that increase the diversification of the solution (using randomness) are very fast. They do not impact the CPU time, and may even deteriorate solution quality on some instances. In our preliminary tests we have performed such analyses in order to fine tune some operators and change those that were too time consuming or that did not have a positive impact on the solution. From this analysis, it seems that operator 9 is the most critical and operator 11 is the least critical.

We have also profiled the code of our heuristic algorithm using `GNU gprof` to identify how the computing time was distributed in the algorithm. To this end, we have solved the subset of instances applied in Tables 3.7−3.9 and found that nearly 50% of the time is spent instantiating and solving network flow problems. Even though this percentage is high, solving network flow problems is still much faster than the alternative, i.e. solving integer linear programs using a general purpose solver. Some functions that are executed at every iteration, sometimes several times, are also time consuming. These include copying solutions (to keep the selected ones and to restore the previous ones over the unaccepted ones), and calculating the cost of a solution. Each of these two functions consumes approximately 10% of the total computing time. The time used by each ALNS operator is roughly the same. This is due to the fact that most of the operators are simple, to the adaptive mechanism inherent to the method, and to the normalization used to the most complex operators, as described in Section 3.5. In general, each ALNS operator uses 1 to 3% of the CPU time over the selected instances.

We also wanted to identify how well some operators would perform if taken in isolation. To this end, we have executed the code with only the few operators we wanted to assess. We offer in Table 3.10 a summary with three different analyses: randomness only, best (worst) insertion (removal) only, and mixed operators only (those that apply removals and insertions simultaneously). These tests were performed on the IRPT-OU on a subset of instances including small, medium and large ones.

Table 3.10: Average increase of the solution cost (%) when only a subset of operators are used

	Randomness only	Best/worst only	Mixed only
Average % increase	0.21	24.58	1.42

Finally, we have conducted experiments on the case with fixed and variable transshipment costs using the model presented in Section 3.3.5. To this end, we have selected 10 instances (all instances with six periods and 15 customers) and we have set various combinations of fixed cost γ and variable cost β, totalling 160 new tests. Table 3.11 presents averages over the 10 instances under an ML policy for the number "#q" of regular deliveries, the average size "Avg q" of the deliveries, the number "#w" of transshipments, the average size "Avg w" of the transshipped quantities, and the running time "Time (s)" in seconds.

Table 3.11: Trade-offs between variable (β) and fixed (γ) transshipment costs and their impact on the quantities delivered

Fixed cost γ	Variable cost β	# q	Avg q	# w	Avg w	Time (s)
0.00	0.00	0.0	0.0	264.7	91.0	0.0
	0.01	27.2	128.5	19.6	31.6	1338.7
	0.10	38.7	99.2	0.7	5.7	1041.3
	1.00	39.2	97.9	0.0	0.0	1660.6
1.00	0.00	0.0	0.0	76.9	107.6	76.1
	0.01	27.2	128.5	16.6	36.6	586.9
	0.10	38.9	98.7	0.5	4.6	627.3
	1.00	39.3	98.1	0.0	0.0	671.1
10.00	0.00	0.0	0.0	48.0	119.6	403.4
	0.01	28.0	126.8	13.2	40.8	1104.6
	0.10	38.9	98.7	0.4	10.5	536.8
	1.00	39.2	97.9	0.0	0.0	90.1
100.00	0.00	9.0	113.9	29.0	105.0	202.9
	0.01	37.1	102.9	1.9	27.5	811.5
	0.10	39.2	98.0	0.0	0.0	147.4
	1.00	39.2	97.9	0.0	0.0	40.0

The first column presents the fixed transshipment cost, with four options for the variable transshipment costs. The topmost case reduces to the problem without any transshipment costs. In this extreme case, all deliveries are made through transshipments. From a computational point of view, this case is the easiest, as reflected by its relatively low running time. As the fixed or variable transshipment cost increases, fewer transshipments are performed, and the quantities transshipped are smaller. The difficulty of the problem does not seem to be directly related to the fixed or variable transshipment costs.

3.7 Conclusions

We have introduced a new variant of the Inventory-Routing Problem, in which planned transshipments are allowed. This problem is very difficult to solve exactly. To generate good solutions, we have developed a branch-and-cut scheme and a powerful ALNS heuristic capable of solving four variants of the problem: IRP-OU, IRPT-OU, IRP-ML and IRPT-ML. Comparative tests on a large set of artificial instances have shown that our heuristic can produce high quality solutions within reasonable computing times. We have also shown that the use of transshipment can reduce solution cost significantly on these instances, depending on the ratio between the unit transshipment cost and the cost of using the supplier's vehicle.

Chapter 4

Consistency in Multi-Vehicle Inventory-Routing

Chapter information

An article based on this chapter was published in *Transportation Research Part C*: L. C. Coelho, J.-F. Cordeau, G. Laporte. Consistency in multi-vehicle inventory-routing. *Transportation Research Part C*, 24(1):270−287, 2012.

An article partly based on the exact algorithm presented in Section 4.3 was published in *Computers & Operations Research*: L. C. Coelho, G. Laporte. Exact Solutions for Several Classes of Inventory-Routing Problems. *Computers & Operations Research*, 40(2):558−565, 2013.

In this chapter we analyze the impact of incorporating regularities to the IRP framework. To this end, we integrate and extend the concept of *consistency* within inventory-routing.

4.1 Introduction

Whereas VMI policies are clearly beneficial from a system's perspective, they may sometimes result in inconveniences both to the supplier and to the customers. This is the case, for example, when very small deliveries take place on consecutive days, followed by a very large delivery, after which the customer is not visited for a long period. Another example, this time undesirable for the supplier, is that it could be optimal to dispatch a mix of almost full and almost empty vehicles, which does not yield a proper load balancing and may irritate some drivers.

Companies need not only provide cost effective solutions to their customers, but also high quality service. This can be partly achieved by incorporating quality of service features in IRP solutions, which should yield a competitive advantage. To this end, we introduce the concept of *consistency* in the IRP in order to reflect some common quality of service standards. This can be achieved, for example, through the application of workforce management policies (Barlett and Ghoshal, 2002; Groër et al., 2009; Smilowitz et al., 2012). Thus, one would expect that regularly assigning the same driver to customers will help create a bond that can benefit both parties. Drivers will gain an increased familiarity with the region and the customer sites assigned to them, and will thus develop a more personal rapport with the customers. Another example of consistency is the spacing of deliveries to customers. To ensure smoother operations, visits should ideally be spread out evenly over the planning horizon. This type of requirement is often modeled as constraints in the context of the periodic Vehicle Routing Problem (VRP) (Christofides and Beasley, 1984; Francis et al., 2008) but it has not yet been imposed in the IRP. Finally, the quantities delivered to customers can also be controlled in order to avoid large variations over time, which are negatively perceived by customers (Beamon, 1999). In this paper, we consider six different consistency features in IRP solutions:

1. Quantity consistency: any delivery performed to a customer must lie within certain customer-dependent intervals, to avoid large variations. From the customers' point of view, delivery size is important. If deliveries are too small, then customers will have to receive frequent visits, which is inconvenient and time-consuming. Deliveries that are too large may create congestion in the warehouse.

2. Vehicle filling rate: a vehicle can only be used if its filling rate lies within a certain interval.

3. Order-up-to (OU) policy: this is a common IRP constraint (see e.g Archetti et al. (2007, 2011, 2012); Bertazzi et al. (2002); Coelho et al. (2012a)) which can be viewed as a consistency feature. It states that whenever a visit is performed to a customer, the delivery should fill the customer's inventory capacity.

4. Driver consistency: this requirement means that each customer is assigned to one driver.

5. Driver partial consistency: one shortcoming of the previous feature is that it

may cause a vehicle to serve very few customers and thus its effect may be very costly. We relax this rule by allowing some deliveries not to be subject to it.

6. Visit spacing: we impose a minimum and a maximum interval between two consecutive visits to the same customer.

Some of these features (e.g. 1 and 6) should depend on the stability of the demand. If the demand is highly variable, customers would expect deliveries to be variable as well, because consistency would then make little sense. However, it is known (Barrat, 2003; Olson and Xie, 2010) that the application of VMI requires some demand stability, which legitimates the consistency features we propose. It is also relevant to note that some of the six consistency features cannot be used in combination with some others. For example, 4 is stronger than 5; the OU policy cannot always be enforced if features 1, 2, or 6 are implemented; other combinations of the consistency features, like 1 and 2, may yield infeasible solution for some parameter values. The choice, application and parameters regulating each consistency feature should be the object of discussion and negotiation between customers and the supplier, as is the case of any VMI strategy (Erhun and Keskinocak, 2011).

The concept of driver consistency has already been applied by Groër et al. (2009) to a version of the VRP in which customers receive visits on prespecified days. The authors have proposed a model ensuring that the same customer is always served by the same driver as a means of improving quality of service, but the application of this constraint to the IRP is new and more complicated because the visit days are endogenous and because of the inventory management issues involved.

We model and solve the *basic* multi-vehicle version of the problem (MIRP) considered in Archetti et al. (2007), Archetti et al. (2012) and Bertazzi et al. (2002) to which we incorporate the consistency features just described. Although the MIRP has previously been studied, the variety of assumptions has left a gap in the literature in the sense that one cannot find benchmarks to a common version of the problem. For instance, to cite some recent contributions to the MIRP literature and their different assumptions, Abdelmaguid and Dessouky (2006) allow backorders and use a non-linear transportation cost function which depends on the quantity delivered, Dauzère-Pérès et al. (2007) have studied the stochastic version of the problem, and Yu et al. (2008) did not include supplier inventory costs. Here we define and solve benchmark instances of the MIRP derived from those of Archetti et al. (2007, 2012) for the single vehicle case, with and without consistency requirements. Our algorithm can also solve the consistent VRP with capacity constraints.

The main scientific contributions of this chapter are to add consistency requirements to the basic MIRP and to develop a branch-and-cut scheme as well as a matheuristic for this version of the MIRP, called the *consistent* MIRP. The remainder of the paper is organized as follows. In Section 4.2 we formally describe the basic MIRP and we present a mixed-integer linear programming formulation for it and for the consistent MIRP. Section 4.3 describes the branch-and-cut algorithm we have developed and Section 4.4 describes our heuristic algorithm which combines adaptive large neighborhood search and the exact solution of mixed integer linear programs. These algorithms can solve the basic MIRP and the consistent MIRP defined by any meaningful combination of the six features just introduced. This is followed by the results of extensive computational experiments in Section 4.5, and by conclusions in Section 4.6.

4.2 Formal problem description and mathematical models

We now formally introduce the basic MIRP. The problem is defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{0, ..., n\}$ is the vertex set and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ is the arc set. Vertex 0 is a depot at which the supplier is located and the vertices of $\mathcal{V}' = \mathcal{V} \setminus \{0\}$ represent customers. The problem is defined over a planning horizon of length p and, at each time period $t \in \mathcal{T} = \{1, ..., p\}$, the quantity of product made available at the supplier is equal to r^t. A unit inventory holding cost h_i is incurred by customer i and by the supplier at each period, and customer i has an inventory holding capacity C_i. We assume the supplier has enough inventory to meet all the demand during the planning horizon and that inventories are not allowed to be negative. The variables I_0^t and I_i^t are defined as the inventory levels at the end of period t, respectively at the supplier and at customer i. At the beginning of the planning horizon the decision maker knows the current inventory level of the supplier and of all customers (I_0^0 and I_i^0 for $i \in \mathcal{V}'$), and has full knowledge of the demand d_i^t of each customer i for each time period t.

A set $\mathcal{K} = \{1, ..., K\}$ of vehicles are available. We denote by Q_k the capacity of vehicle k. Each vehicle is able to perform one route per time period, from the supplier to a subset of customers. A routing cost c_{ij} is associated with arc $(i, j) \in \mathcal{A}$.

The objective of the problem is to minimize the total routing and inventory holding cost while meeting the demand for each customer. The replenishment plan is subject to the following constraints:

- at the end of period t, the inventory at a customer location cannot exceed its maximum capacity;

- inventories are not allowed to be negative;
- the supplier's vehicles can each perform at most one route per time period;
- each route starts and ends at the depot;
- the vehicle capacities cannot be exceeded.

The solution to the problem specifies which customers to serve at each time period, which vehicle to use on each route, how much to deliver to each visited customer, and how to sequence customers on the vehicle routes. Throughout the paper, we assume that the quantity r^t becoming available at the supplier in period t can be used for deliveries to customers in the same period, and that the quantities q_i^{kt} received by customer i in period t can be used to meet the demand in that period.

Our model belongs to the same family as those of Bertazzi et al. (2002) and Archetti et al. (2007, 2012) for the single vehicle IRP and of Coelho et al. (2012a) for the single vehicle IRPT. It works with the following binary variables: x_{ij}^{kt} is equal to 1 if and only if vertex j immediately follows vertex i on the route of vehicle k in period t, and y_i^{kt} is equal to 1 if and only if customer i is visited by vehicle k in period t. We denote by q_i^{kt} the quantity of product delivered from the supplier to customer i using vehicle k in time period t. The model also uses continuous variables w_i^{kt} to enforce the VRP subtour elimination constraints (Desrochers and Laporte, 1991; Kara et al., 2004). They represent the sum of the deliveries made by vehicle k in period t after visiting customer i.

4.2.1 Mixed integer linear program for the basic MIRP

The mathematical model for the basic MIRP is as follows:

$$\text{(MIRP)} \quad \text{minimize} \sum_{t \in \mathcal{T}} h_0 I_0^t + \sum_{i \in \mathcal{V}'} \sum_{t \in \mathcal{T}} h_i I_i^t + \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} c_{ij} x_{ij}^{kt} \tag{4.1}$$

subject to

$$I_0^t = I_0^{t-1} + r^t - \sum_{i \in \mathcal{V}'} \sum_{k \in \mathcal{K}} q_i^{kt} \qquad t \in \mathcal{T} \tag{4.2}$$

$$I_0^t \geq 0 \qquad t \in \mathcal{T} \tag{4.3}$$

$$I_i^t = I_i^{t-1} + \sum_{k \in \mathcal{K}} q_i^{kt} - d_i^t \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.4}$$

$$I_i^t \geq 0 \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.5}$$

$$I_i^t \leq C_i \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.6}$$

$$\sum_{k \in \mathcal{K}} q_i^{kt} \leq C_i - I_i^{t-1} \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.7}$$

$$\sum_{k \in \mathcal{K}} q_i^{kt} \leq C_i \sum_{j \in \mathcal{V}} \sum_{k \in \mathcal{K}} x_{ij}^{kt} \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.8}$$

$$\sum_{i \in \mathcal{V}'} q_i^{kt} \leq Q_k \qquad t \in \mathcal{T}, k \in \mathcal{K} \tag{4.9}$$

$$q_i^{kt} \leq y_i^{kt} C_i \qquad i \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K} \tag{4.10}$$

$$\sum_{j \in \mathcal{V}} x_{ij}^{kt} = \sum_{j \in \mathcal{V}} x_{ji}^{kt} = y_i^{kt} \qquad i \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K} \tag{4.11}$$

$$\sum_{j \in \mathcal{V}'} x_{0j}^{kt} \leq 1 \qquad k \in \mathcal{K} \quad t \in \mathcal{T} \tag{4.12}$$

$$\sum_{k \in \mathcal{K}} y_i^{kt} \leq 1 \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.13}$$

$$w_i^{kt} - w_j^{kt} + Q_k x_{ij}^{kt} \leq Q_k - q_j^{kt} \qquad i \in \mathcal{V}', j \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K} \tag{4.14}$$

$$q_i^{kt} \leq w_i^{kt} \leq Q_k \qquad i \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K} \tag{4.15}$$

$$q_i^{kt} \geq 0 \qquad i \in \mathcal{V}', j \in \mathcal{V}, t \in \mathcal{T}, k \in \mathcal{K} \tag{4.16}$$

$$x_{ij}^{kt}, y_i^{kt} \in \{0,1\} \qquad i,j \in \mathcal{V}, i \neq j, t \in \mathcal{T}, k \in \mathcal{K}. \tag{4.17}$$

In this model, the objective function is the sum of inventory costs at the supplier and customer locations, and of routing costs. Constraints (4.2) define the inventory at the supplier carried at the end of period t. Constraints (4.3) forbid stockouts at the supplier. Constraints (4.4) and (4.5) are similar to (4.2) and (4.3) but apply to the customers. Constraints (4.6) define the maximum inventory level at customer locations, while constraints (4.7) and (4.8) ensure that the quantity delivered to customer i at period t will not exceed the customer's inventory capacity if the customer is served, and will be zero otherwise. Constraints (4.9) mean that vehicle capacities are never exceeded. Constraints (4.10)−(4.15) impose linking and routing conditions. In particular, constraints (4.14) ensure the consistency of the load of each vehicle along its route and prevent subtours. Finally, constraints (4.16) and (4.17) enforce the non-negativity and integrality requirements.

4.2.2 Modeling the features of the consistent MIRP

We now formally describe the features of six versions of the consistent MIRP and we show how they can be modeled separately or jointly.

4.2.2.1 Quantity consistency

A way to ensure that all deliveries to a given customer will be consistent over time is to force the delivery amounts to lie within an interval $[g_l,\, g_u]$ around a target value equal to the average demand of the customer over the planning horizon:

$$y_i^{kt} g_l \sum_{t \in \mathcal{T}} d_i^t/p \leq q_i^{kt} \leq y_i^{kt} g_u \sum_{t \in \mathcal{T}} d_i^t/p \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{4.18}$$

4.2.2.2 Vehicle filling rate

To balance the load between vehicles and to avoid dispatching vehicles with very low loads, we impose a vehicle filling rate constraint which specifies that a vehicle can only be used if the total quantity it delivers fills at least a fraction γ of its capacity. This is achieved by adding the following constraint to the basic model:

$$\sum_{i \in \mathcal{V}'} q_i^{kt} \geq \gamma \sum_{i \in \mathcal{V}'} x_{0i}^{kt} Q_k \qquad k \in \mathcal{K}, t \in \mathcal{T}. \tag{4.19}$$

4.2.2.3 Order-up-to policy

Under an OU inventory policy, the decisions of when and how much to deliver to a customer are linked: whenever a customer is visited, the quantity delivered must fill the customer's inventory capacity. The OU policy is imposed through the constraints

$$q_i^{kt} \geq C_i \sum_{j \in \mathcal{V}} x_{ij}^{kt} - I_i^{t-1} \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{4.20}$$

4.2.2.4 Driver consistency

Driver consistency is modeled with an extra binary variable z_i^k equal to 1 if and only if vehicle k visits customer i. Then, three sets of constraints are added to the basic model:

$$\sum_{k \in \mathcal{K}} z_i^k = 1 \qquad i \in \mathcal{V}' \tag{4.21}$$

$$y_i^{kt} \leq z_i^k \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.22}$$

$$z_i^k \in \{0, 1\} \qquad i \in \mathcal{V}', k \in \mathcal{K}. \tag{4.23}$$

Constraints (4.21) ensure that exactly one vehicle is assigned to each customer over the planning horizon. Constraints (4.22) allow deliveries only from the vehicle assigned to the customer.

4.2.2.5 Driver partial consistency

It may sometimes be preferable to apply a partial consistency policy by which a large number of deliveries follow the driver consistency rule, but in some cases the rule may be relaxed. We have modeled this policy by adding to the objective function a penalty term proportional to the number of extra vehicles assigned to each customer, beyond their regular vehicle. We have introduced a binary variable s_i^k indicating whether an extra vehicle k is assigned to customer i, and we impose the following sets of constraints to the basic model:

$$\sum_{k \in \mathcal{K}} z_i^k = 1 \qquad i \in \mathcal{V}' \tag{4.24}$$

$$y_i^{kt} \leq z_i^k + s_i^k \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.25}$$

$$s_i^k, z_i^k \in \{0,1\} \qquad i \in \mathcal{V}', k \in \mathcal{K}. \tag{4.26}$$

Constraints (4.24) assign a first vehicle to each customer, while constraints (4.25) allow additional vehicles to be assigned to the same customer. We then add a penalty term

$$\alpha \sum_{i \in \mathcal{V}'} \sum_{k \in \mathcal{K}} s_i^k \tag{4.27}$$

to the objective function (4.1). By adjusting the parameter α, one can control how restrictive the driver partial consistency policy will be.

4.2.2.6 Visit spacing

One may also want to enforce a minimum and maximum time interval between two consecutive visits to the same customer, since it may be undesirable to visit the same customer on several successive days or to leave a customer unvisited for a long period. Adding the following constraints to the basic model will ensure that at least one visit will take place every $(M_i + 1)$ periods, and no more than one visit will take place in any $(m_i + 1)$ successive periods:

$$\sum_{k \in \mathcal{K}} \sum_{l=t}^{t+m_i} y_i^{kl} \leq 1 \qquad i \in \mathcal{V}', t \in \{1, ..., p - m_i\} \tag{4.28}$$

$$\sum_{k \in \mathcal{K}} \sum_{l=t}^{t+M_i} y_i^{kl} \geq 1 \qquad i \in \mathcal{V}', t \in \{1, ..., p - M_i\}. \tag{4.29}$$

In practice, both M_i and m_i should depend on the capacity and on the demand of customer i.

4.3 A branch-and-cut algorithm

We have developed a branch-and-cut algorithm which works on top of an undirected formulation of the problem as in the previous chapter. This formulation is designed to consider multiple vehicles. In case the fleet is homogeneous, it yields much symmetry. We tighten this formulation by imposing the following symmetry breaking constraints valid for the case where the vehicle fleet is homogeneous:

$$y_0^{kt} \leq y_0^{k-1,t} \quad k \in \mathcal{K} \backslash \{1\} \quad t \in \mathcal{T} \tag{4.30}$$

$$y_i^{kt} \leq \sum_{j<i} y_j^{k-1,t} \quad i \in \mathcal{V} \quad k \in \mathcal{K} \backslash \{1\} \quad t \in \mathcal{T}. \tag{4.31}$$

Constraints (4.30) ensure that vehicle k cannot leave the depot if vehicle $k-1$ is not used. This symmetry breaking rule is then extended to the customer vertices by constraints (4.31) which state that if customer i is assigned to vehicle k in period t, then vehicle $k-1$ must serve a customer with an index smaller than i in the same period. These constraints are inspired from those proposed by Fischetti et al. (1995) for the capacitated vehicle routing problem and by Albareda-Sambola et al. (2011) for a plant location problem.

This formulation can handle the basic MIRP as well as all the consistency features as described in Section 4.2.2. Note that the basic MIRP has been recently solved by means of a similar branch-and-cut algorithm by Adulyasak et al. (2012) as a special case of the Production-Routing Problem (PRP).

4.4 A matheuristic for the consistent MIRP

The MIRP is $\mathcal{NP}$-hard since it generalizes the capacitated VRP. As a result, the models described in Section 4.2 can only be used for the exact solution of relatively small and medium size instances. For this reason, we have opted to solve the problem heuristically. The heuristic we have developed can solve the basic MIRP and any meaningful combination of the six versions of the consistent MIRP just defined. It applies an Adaptive Large Neighborhood Search (ALNS) scheme in which some subproblems are solved exactly as MILPs. It can therefore be described as a *matheuristic* (Maniezzo et al., 2009), i.e. as a hybridization of a heuristic and of a mathematical programming algorithm. The concept of ALNS was put forward by Ropke and Pisinger (2006a) in the context of the capacitated VRP. It has since be successfully applied to several related problems such as the vehicle scheduling problem (Bartodziej et al., 2009; Pepin et al., 2009), the fixed charged network flow problem (Hewitt et al., 2010), the stochastic arc routing problem (Laporte et al., 2010) and several classes of vehicle routing problems (Ropke and Pisinger, 2006b). Matheuristics have already been applied to other types of vehicle routing including the works of Prins et al. (2007), Tarantilis et al. (2009) and Wolfler Calvo and Touati-Moungla (2011).

4.4.1 Adaptive Large Neighborhood Search

Our ALNS heuristic follows the general framework proposed by Ropke and Pisinger (2006a) and works as follows. At each iteration, a number of customers are removed from their current route by a destroy operator and are eventually reinserted back elsewhere by a repair operator. The choice of an operator is governed by a roulette-wheel mechanism. Each operator i is assigned a weight ω_i whose value depends on its past performance, as well as a score. Given h operators with weights ω_i, operator j will be selected with probability $\omega_j / \sum\limits_{i=1}^{h} \omega_i$. Initially, all weights are equal to one and all scores are equal to zero. At each iteration, the score of the selected operator is increased by σ_1 if it finds a new best solution, by σ_2 if it finds a solution better than the incumbent, and by σ_3 if the solution is not better but is still accepted. Obviously $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$. The search is divided into segments of φ iterations each, after which the weights and scores are updated as follows. Let π_i and o_{ij} be, respectively, the score of operator i and the number of times it has been used in the last segment j, normalized by a factor $\nu_i \geq 1$ reflecting the computational effort it requires (see Coelho et al. (2012a); Ropke and Pisinger (2006b)). The *normalization factor* ν_i multiplies o_{ij}, and therefore decreases the weight of operator i, so that the more time consuming operators are applied less frequently. The values used for the normalization factors are all equal to one in our implementation, except for two cases where different values are used. These are provided in Sections 3.1.1 and 3.1.2. The updated weights are then

$$\omega_i := \begin{cases} \omega_i & \text{if } o_{ij} = 0 \\ (1-\eta)\omega_i + \eta \pi_i / \nu_i o_{ij} & \text{if } o_{ij} \neq 0, \end{cases} \tag{4.32}$$

where $\eta \in [0, 1]$ is called the reaction factor, controlling how quickly the weight adjustment reacts to changes in the movement performance (see Section 4.4.3). All scores are reset to zero.

As in Ropke and Pisinger (2006b) we use the same acceptance criterion as in simulated annealing: given a solution s, a neighbor solution s' is accepted if $z(s') < z(s)$, and with probability $e^{-(z(s')-z(s))/\tau}$ otherwise, where $z(s)$ is the solution cost and $\tau > 0$ is the current temperature. The temperature is initialized at τ_{start} and is decreased by a cooling rate factor ϕ at each iteration, where $0 < \phi < 1$.

Our computational tests have shown that the initial solution does not have a significant impact on the overall solution cost or on the running time. We therefore initialize the search with a randomly generated solution by assigning a random number of customers to random periods and vehicles. This initial solution is not necessarily feasible. Our algorithm also works if the initial solution is empty, in which case the destroy operators do not initially apply.

4.4.1.1 Destroy operators

1. **Randomly remove $\boldsymbol{\rho}$**: This operator randomly selects one period and one vehicle and removes one randomly selected customer from it. It is repeated ρ times. The operator is useful for refining the solution, since it does not change it much when ρ is small (which happens frequently), but still yields a major transformation when ρ is large.

2. **Remove worst $\boldsymbol{\rho}$**: This operator removes the customer that will save the most when removed, considering the total routing and inventory cost. It is applied ρ times. Its normalization factor is 20.

3. **Shaw removal**: Following the ideas developed in Ropke and Pisinger (2006a) and Shaw (1997), this operator removes customers that are relatively close to each other. Specifically, it randomly selects one vehicle, one period and one customer served in this period, it computes the distance $dist_{min}$ to the closest customer also being served by the same route, and it removes all customers within $2dist_{min}$ units from the selected route.

4. **Avoid consecutive visits**: This operator is based on our observation that good solutions often do not contain visits to the same customer on two consecutive periods. Then, the operator verifies whether any customer is visited on two consecutive periods and removes the latest visit.

5. **Empty one period**: This operator selects one random period and empties all routes performed during that period.

6. **Empty one vehicle**: This operator selects one random vehicle and empties all routes performed by this vehicle.

4.4.1.2 Repair operators

1. **Randomly insert $\boldsymbol{\rho}$**: This operator randomly inserts ρ customers into the current solution. Specifically, it selects one random customer, one random vehicle and one random period, and inserts the customer into the route of that vehicle in that period if it is not already routed in the same period. This operator is applied ρ times.

2. **Insert best $\boldsymbol{\rho}$**: This operator is analogous to the previous one. It is applied ρ times by computing the cheapest insertion with respect to the total cost. The normalization factor used for this operator is 20.

3. **Shaw insertions**: This operator is similar to the Shaw removal operator in the sense that it selects similar customers to be inserted together. It selects one vehicle, one period and one customer not served in that period by any vehicle. The operator then computes $dist_{min}$ and all customers within a $2dist_{min}$ distance are inserted in the same route, always following the cheapest insertion rule.

4. **Swap $\boldsymbol{\rho}$ customers**: This operator selects two customers from two different routes and swaps their assignments, following the cheapest insertion rule. It is also applied ρ times.

4.4.2 Exact subproblem solutions

Our matheuristic embeds the exact solution of two subproblems. The first one, called Delivery Quantities (DQ), optimizes the delivery quantities associated with a given set of vehicle routes. It is solved every time a new routing solution is computed by the ALNS mechanism. It uses a binary parameter $\bar{x}_{ij}^{kt}$ equal to one if and only if customer j follows customer i in the route of vehicle k in period t. As shown in Coelho et al. (2012a), DQ can be formulated as the following network flow problem:

$$\text{(DQ)} \quad \text{minimize} \sum_{t \in \mathcal{T}} h_0 I_0^t + \sum_{i \in \mathcal{V}'} \sum_{t \in \mathcal{T}} h_i I_i^t \tag{4.33}$$

subject to

$$I_0^t = I_0^{t-1} + r^t - \sum_{i \in \mathcal{V}'} \sum_{k \in \mathcal{K}} q_i^{k,t} \qquad t \in \mathcal{T} \tag{4.34}$$

$$I_i^t = I_i^{t-1} + \sum_{k \in \mathcal{K}} q_i^{k,t} - d_i^t \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.35}$$

$$I_0^t \geq 0 \qquad t \in \mathcal{T} \tag{4.36}$$

$$I_i^t \geq 0 \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.37}$$

$$I_i^t \leq C_i \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.38}$$

$$\sum_{k \in \mathcal{K}} q_i^{kt} \leq C_i - I_i^{t-1} \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.39}$$

$$\sum_{k \in \mathcal{K}} q_i^{kt} \leq C_i \sum_{j \in \mathcal{V}} \sum_{k \in \mathcal{K}} \bar{x}_{ij}^{kt} \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{4.40}$$

$$\sum_{i \in \mathcal{V}'} q_i^{kt} \leq Q_k \qquad t \in \mathcal{T}, k \in \mathcal{K}. \tag{4.41}$$

Constraints (4.34) and (4.35) define the flow conservation conditions. Lower and upper bounds on the flows are defined by (4.36)−(4.40). Vehicle capacity constraints (4.41) still define an upper bound on the quantity delivered by the vehicle, even though the customers to be visited are now fixed.

The goal of the second subproblem, called Solution Improvement (SI), is to find the best solution that results from removal and reinsertion operations to a given solution and to evaluate its cost. To limit computational effort, the number of operations is limited, and approximate removal and insertion costs are used. This problem is no longer a network flow problem. It is solved every θ iterations or whenever a new best solution has been identified. Using an idea proposed by Archetti et al. (2012), we simplify and approximate the routing costs resulting from vertex removals and reinsertions as follows. Let a_i^{kt} represent the routing cost reduction if customer i is removed from the route of vehicle k at period t, which obviously visits customer i; let b_i^{kt} represent the routing cost if customer i is inserted in the route of vehicle k at period t, which obviously does not already visit customer i; finally, let r_i^{kt} be a binary parameter equal to 1 if and only if customer i is visited in the current route of vehicle k at period t. Also define the following binary variables: let u_i^{kt} be equal to 1 if and only if customer i is removed from the existing route of vehicle k at period t, and let v_i^{kt} be equal to 1 if and only if customer i is inserted in the route of vehicle k at period t. This subproblem is then to

$$\text{(SI)} \quad \text{minimize} \sum_{t \in \mathcal{T}} h_0 I_0^t + \sum_{i \in \mathcal{V}'} \sum_{t \in \mathcal{T}} h_i I_i^t - \sum_{i \in \mathcal{V}'} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} a_i^{kt} u_i^{kt} + \sum_{i \in \mathcal{V}'} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} b_i^{kt} v_i^{kt} \tag{4.42}$$

subject to (2)−(6) and

$$q_i^{kt} \leq C_i - I_i^{t-1} \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.43}$$

$$q_i^{kt} \leq (r_i^{kt} - u_i^{kt} + v_i^{kt}) C_i \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.44}$$

$$v_i^{kt} \leq 1 - r_i^{kt} \quad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.45}$$

$$u_i^{kt} \leq r_i^{kt} \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.46}$$

$$\sum_{i \in \mathcal{V}'} u_i^{kt} + \sum_{i \in \mathcal{V}'} v_i^{kt} \leq \beta \qquad k \in \mathcal{K}, t \in \mathcal{T} \tag{4.47}$$

$$\sum_{i \in \mathcal{V}'} q_i^{kt} \leq Q_k \qquad k \in \mathcal{K}, t \in \mathcal{T} \tag{4.48}$$

$$q_i^{kt} \geq 0 \qquad i \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K} \tag{4.49}$$

$$u_i^{kt}, v_i^{kt} \in \{0, 1\} \qquad i \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K}. \tag{4.50}$$

The objective function (4.42) minimizes the total inventory, removal and insertion cost. Constraints (4.43)−(4.44) are similar to (7)−(8) and enforce the ML policy. Constraints (4.45) ensure that if a customer is already present in a route, it cannot be reinserted in the same route. Likewise, constraints (4.46) guarantee that only those customers belonging to a route can be removed from it. Constraints (4.48) ensure that vehicle capacities are respected. If the incumbent solution is changed by more than one customer, then this model only provides an approximation of the actual routing costs. For this reason, we have decided to limit the number of insertions and removals that could take place in the solution of SI, and we have added constraints (4.47) to limit the number of insertions and removals per route to a small value β. Unlike a destroy and repair mechanism which usually means that decisions are made in two successive steps, SI removes and reinserts vertices by taking both decisions at the same time.

4.4.2.1 Quantity consistency

Guaranteeing a minimum and a maximum delivery quantity to each customer is controlled by adding the following constraints to SI, which ensures that the quantities delivered lie within their specified intervals:

$$q_i^{kt} \geq (r_i^{kt} - u_i^{kt} + v_i^{kt}) g_l \sum_{t \in \mathcal{T}} d_i^t / p \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.51}$$

$$q_i^{kt} \leq (r_i^{kt} - u_i^{kt} + v_i^{kt}) g_u \sum_{t \in \mathcal{T}} d_i^t / p \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{4.52}$$

4.4.2.2 Vehicle filling rate

To ensure a minimum vehicle filling rate in SI, the following constraints are added. They use new binary variables y^{kt} equal to 1 if and only if vehicle k is used in period t:

$$y^{kt} \geq z_i^{kt} \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.53}$$

$$\sum_{i \in \mathcal{V}'} q_i^{kt} \geq \gamma y^{kt} Q_k \qquad k \in \mathcal{K}, t \in \mathcal{T} \tag{4.54}$$

$$y^{kt} \in \{0, 1\} \qquad k \in \mathcal{K}, t \in \mathcal{T}. \tag{4.55}$$

4.4.2.3 Order-up-to policy

The OU policy is handled through the following constraints:

$$q_i^{kt} \geq (r_i^{kt} - u_i^{kt} + v_i^{kt})C_i - I_i^{t-1} \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{4.56}$$

These constraints ensure that if a delivery to a customer is performed, the quantity delivered should be at least equal to the difference between its current inventory and its inventory holding capacity. Together with constraints (4.43) and (4.44), they ensure that the quantity delivered will exactly fill the customer's inventory capacity.

4.4.2.4 Driver consistency

The driver consistency requirement is modeled in SI by means of an extra binary variable z_i^k equal to 1 if and only if vehicle k visits customer i, as it was defined in Section 4.2.2.4. Then, three sets of constraints are added to the SI model:

$$\sum_{k \in \mathcal{K}} z_i^k = 1 \qquad i \in \mathcal{V}', k \in \mathcal{K} \tag{4.57}$$

$$r_i^{kt} - u_i^{kt} + v_i^{kt} \leq z_i^k \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.58}$$

$$z_i^k \in \{0, 1\} \qquad i \in \mathcal{V}', k \in \mathcal{K}. \tag{4.59}$$

Constraints (4.57) ensure that exactly one vehicle is assigned to each customer, while constraints (4.58) only allow deliveries from the vehicle assigned to that customer.

4.4.2.5 Driver partial consistency

The driver partial consistency is also modeled in SI with a binary variable s_i^k and a penalty in the objective function, as above. The variable s_i^k will be equal to one if and only if an extra vehicle k is assigned to customer i. The required constraints are

$$\sum_{k \in \mathcal{K}} z_i^k = 1 \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.60}$$

$$r_i^{kt} - u_i^{kt} + v_i^{kt} \leq z_i^k + s_i^k \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{4.61}$$

$$s_i^k, z_i^k \in \{0,1\} \qquad i \in \mathcal{V}', k \in \mathcal{K}. \tag{4.62}$$

The penalty to the objective function is added in the same fashion as in Section 4.2.2.5.

4.4.2.6 Visit spacing

The imposition of minimum and maximum intervals between visits is modeled by adding the following sets of constraints to the SI model:

$$\sum_{k\in\mathcal{K}} \sum_{l=t}^{t+m_i} (r_i^{kr} - u_i^{kl} + v_i^{kr}) \leq 1 \qquad i \in \mathcal{V}', t \in \{1, ..., p - m_i\} \tag{4.63}$$

$$\sum_{k\in\mathcal{K}} \sum_{l=t}^{t+M_i} (r_i^{kr} - u_i^{kl} + v_i^{kr}) \geq 1 \qquad i \in \mathcal{V}', t \in \{1, ..., p - M_i\}. \tag{4.64}$$

4.4.3 Parameter settings

We now describe the parameters that govern our algorithm. We have tested different combinations for the parameters during a tuning phase. We have evaluated how the algorithm performed with different numbers of iterations. To this end, we have run it 5,000, 10,000, 15,000, 20,000, 25,000, 30,000, 40,000 and 50,000 iterations on a small subset of instances. We then computed the average solution gap that each number of iterations provided with respect to the best solution found. Since the drop of the average gap is steep when the algorithm reaches 50,000 iterations and only equal to 0.12% we have decided to run the algorithm for 50,000 iterations without a time limit. Figure 4.1 depicts the performance just described.

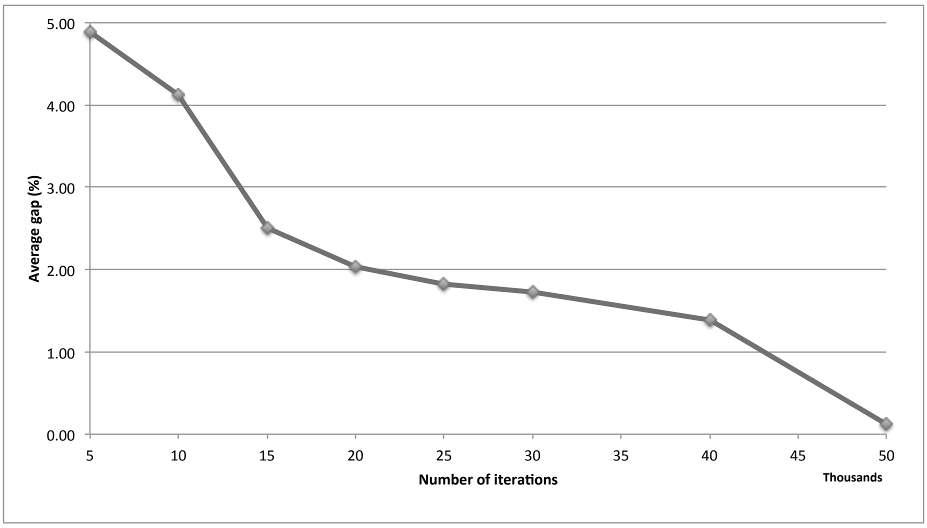

Figure 4.1: Average solution gap over different number of iterations.

The starting temperature τ_{start} is set to 30,000 and the cooling rate ϕ is 0.999701, which yields roughly 50,000 iterations. The stopping criterion is satisfied when the temperature reaches 0.01 or when 50,000 iterations have been performed. We have decided not to stop the algorithm after a predetermined running time because we wanted to evaluate the impact of the different policies themselves, not an algorithmic performance. The segment length φ was set to 200 iterations and the reaction factor η was set to 0.8, that is, new weights will reflect 80% of the performance of the last segment and 20% of the last weight value. Scores are updated with $\sigma_1 = 10$, $\sigma_2 = 5$ and $\sigma_3 = 2$. A trade-off must be made between the CPU consumption and the quality of each operator of the ALNS, as well as how often SI is solved. We have evaluated this trade-off and decided to solve this subproblem with $\beta = 10$ every $\theta = 40$ ALNS iterations, which proved to be a good compromise between computing time and solution quality.

4.4.4 Special rules

The algorithm can handle all six consistency features without modifications. However, its performance can be improved if some adjustments are made to better handle some features.

The first adjustment consists in applying the *avoid consecutive visits* operator only to the basic MIRP, since it could conflict with some of the consistency features proposed, thus decreasing the effectiveness of the algorithm. For example, it may pay to visit some customers on two consecutive periods if this helps achieve a better vehicle filling rate. Similarly, a later visit to a customer can be anticipated if this reduces routing costs (due to geographical proximity) or if this improves driver consistency. After some tests and considerations, we realized that whenever this operator is applied, it directs the search towards good neighborhoods, leading to better solutions. The idea is that a good solution should not visit the same customer on consecutive days, considering that it usually has sufficient inventory to meet its demand and that the number of vehicles and their capacity are limited, and their use is expensive. We have evaluated the impact of the *avoid consecutive visits* operator during the search, by running the algorithm on a subset of instances, both with and without this operator. The results of this experiment are depicted in Figure 4.2. It is clear from Figure 4.2 that the operator has a positive impact on the search process. The average percentage gap with respect to the best solution value found in this experiment is always smaller when the operator is applied. This operator is a direct result of the *visit spacing* consistency feature. We have tried different ideas from other consistency features, but none proved to be as effective for the general case.

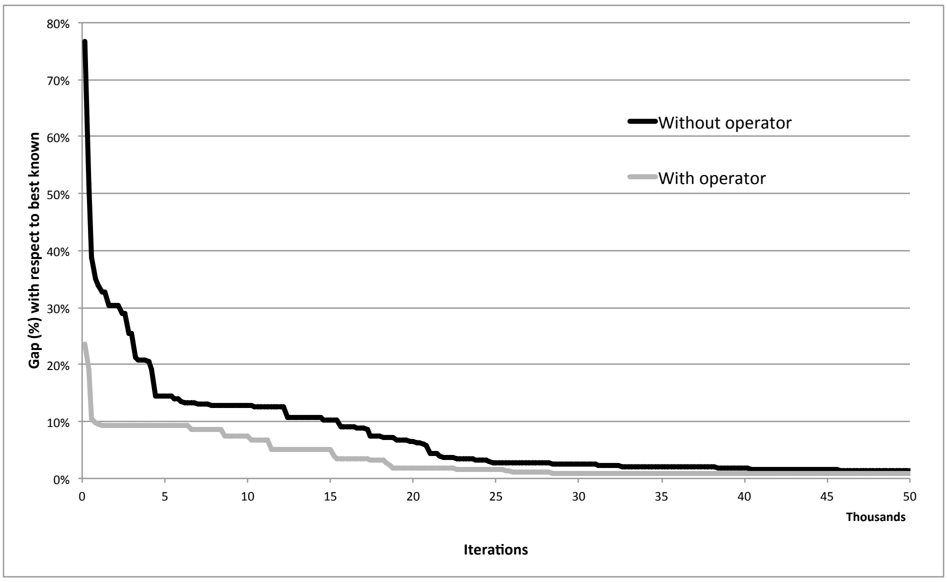

Figure 4.2: Impact of the *avoid consecutive visits* operator.

The second modification relates to implementation details of the different consistency features proposed. For some variants of the main problem, we have made slight modifications to the ALNS operators and to the associated network flow model in order to take into account the specifics of the variant under consideration. In order to enforce the driver consistency rule, we have modified the ALNS operators to allow insertions of customers only in vehicles that had already visited them earlier in the current solution. For the driver partial consistency rule, the only modification needed was related to the computation of the solution cost, in order to take into account the number of vehicles assigned to each customer. For the visit spacing case, the only modifications were made to the insertion operators of the ALNS, as was the case for the driver consistency feature. The OU policy was modeled directly into the remaining network flow problem as in Coelho et al. (2012a), as were the minimum and maximum delivered quantity in the quantity consistency requirements. For the vehicle filling rate case, we have opted to solve SI after each ALNS iteration to help regain feasibility since in this case many ALNS operations yield infeasible solutions.

The third adjustment concerns the SI subproblem. Since it provides an approximation of the true routing costs, it is possible that after applying it to a solution, the output has a higher solution cost than the input. For this reason, we only accept the SI solution if it is better than the solution to which it was applied.

4.4.5 Summary of the algorithm

Algorithm 4.1 provides the pseudocode of our matheuristic.

4.5 Computational experiments

The matheuristic algorithm just described was coded in C++. We have used the scaling push-relabel algorithm developed by Goldberg (1997) for the minimum-cost flow problem to solve DQ, and IBM Concert Technology and CPLEX 12.2 as the solver for SI. Computations were executed on a grid of Dual Core AMD Opteron(tm) Processor 275 machines running at 2.20 GHz, each with 12 GB of RAM installed, running a Linux operating system. The branch-and-cut algorithm was coded in C++ using IBM Concert Technology and CPLEX 12.3 with six threads. Computations were executed on a grid of Intel Xeon™ processors running at 2.66 GHz with up to 48 GB of RAM installed per node, with the Scientific Linux 6.1 operating system.

To evaluate the performance of the algorithms, we have adapted to the multi-vehicle case the 160 small single vehicle IRP instances of Archetti et al. (2007, 2012). These were used in Archetti et al. (2012); Bertazzi et al. (2002); Coelho et al. (2012a) to evaluate single vehicle algorithms for the IRP and are made up of instances with up to three time periods and 50 customers, and six time periods and 30 customers. These instances are described as small-n-low or small-n-high, where the last field refers to a low or high inventory holding cost. There are five instances for each combination and we report average statistics over these. The second set is more recent and contains 60 larger instances proposed in Archetti et al. (2012), with up to six time periods and 200 customers. They are described as large-n-low or large-n-high. There are 10 instances for each combination and we again report average values. We have adapted these instances to account for multiple vehicles by dividing the original vehicle capacity by the number of vehicles considered. Whereas our formulation and algorithm can handle heterogeneous fleet, all our tests are conducted with a homogeneous fleet, which reduces the number of parameters to consider. We have tested our algorithm on the smaller set with two and three vehicles, and on the larger set with two to five vehicles. In total, we have solved $160 \times 2 + 60 \times 4 = 560$ instances for the basic MIRP. In the case of the consistent MIRP, we have solved instances with three vehicles. Since we have defined six versions of this problem, this means that an additional $6 \times (160 + 60) =$ 1,320 instances were solved. For full results on all instances the reader is referred to Coelho et al. (2011b) and to Appendix A.2 for heuristic solutions, and to Appendix A.3 for exact solution values.

Algorithm 4.1 Matheuristic pseudocode

```
Initialize weights of removal and insertion operators to 1 and scores to 0.
$s_{best} \leftarrow s \leftarrow$ *initial solution*.
$\tau \leftarrow \tau_{start}$.
while $\tau > 0.01$ and *iterations* $< 50{,}000$ do
  $s' \leftarrow s$.
  Select a destroy and a repair operator using the roulette-wheel and apply it to $s'$.
  Fix routing decisions, solve DQ to determine the delivery quantities.
  if $f(s') < f(s)$ then
    $s \leftarrow s'$;
    if $f(s) < f(s_{best})$ then
      Solve the SI model associated with $s$;
      $s_{best} \leftarrow s$;
      increase the score of the operators by $\sigma_1$;
    else
      increase the score of the operators by $\sigma_2$;
    end if
  else
    if $s'$ is accepted by the simulated annealing criterion then
      $s \leftarrow s'$;
      increase the score of the operators by $\sigma_3$.
    end if
  end if
  if the iteration count is a multiple of $\varphi$ then
    update the weights of all operators and reset their scores.
  end if
  if the iteration count is a multiple of $\theta$ then
    solve the SI model associated with $s$.
  end if
end while

return $s_{best}$;
```

4.5.1 Stability test

We have first analyzed the stability of the heuristic algorithm by running the same instance five times and then calculating its *coefficient of variation* (CV) which is a normalized and dimensionless measure of dispersion of a probability distribution. The lower the value of CV, the more stable the algorithm is. The coefficient of variation is computed as $CV = S/\bar{X}$, where $\bar{X}$ is the sample average and S is the sample standard deviation.

For each of eight variants of the MIRP and 16 combinations of n and p, we have selected an instance and solved it five times, yielding a total of 640 runs. For each instance we have computed the CV over the five runs. The eight variants and the average CV values of the 16 instances of each variant are reported in Table 4.1. Results indicate that the average CV values are very small (typically no more than 0.01), which is a strong indication of the stability of our algorithm. Given this, we feel justified to run each instance only once in the subsequent tests.

Table 4.1: Average coefficients of variations over five runs of 16 instances for each of eight MIRP variants

Scenario	CV
Basic MIRP, $K = 2$	0.01
Basic MIRP, $K = 3$	0.01
Quantity consistency, $K = 3$	0.02
Vehicle filling rate, $K = 3$	0.01
Order-up-to, $K = 3$	0.01
Driver consistency, $K = 3$	0.01
Driver partial consistency, $K = 3$	0.01
Visit spacing, $K = 3$	0.00
Average	0.01

4.5.2 Computational experiments for the basic MIRP

We have then run our algorithms on a special case of the MIRP with only one vehicle ($K = 1$). This problem has already been solved exactly by means of a branch-and-cut algorithm by Archetti et al. (2007) and is a good starting point to evaluate the performance of our heuristic. Our branch-and-cut could find all optimal solutions in very short running time. We provide in Table 4.2 a comparison between the optimal solutions and the solutions provided by our heuristic for this case. In this table the first column shows the name of the instance; the second presents the average of the optimal solutions obtained in Archetti et al. (2007); the last three columns show the average solution obtained by our heuristic, the percentage gap with respect to the optimal one and the running time of our algorithm in seconds. As can be seen, our heuristics yields quasi-optimal solutions on most instances, and the average optimality gap over 160 instances is only 0.37%.

We also provide in Tables 4.3 and 4.4 the average optimal solution values yielded by our branch-and-cut and by our ALNS-based heuristic over the five small basic MIRP instances with two and three vehicles for $p = 3$ and $p = 6$, respectively. Gaps are reported with respect to the best known lower bound. Note that running times cannot be directly compared as the algorithms were run on different machines. However, as expected the running time for large instances is very high for the exact algorithm, while it remains acceptable for the heuristic one. In addition, we have run our ALNS heuristic over the 10 large basic MIRP instances with two to five vehicles, and $p = 6$. Table 4.5 contains average solution values for each size. For relatively small size instances, our exact method is able to find good solutions. We present in Table 4.6 computational results of our branch-and-cut algorithm on a subset of these instances.

In a second stream of experiments, we have studied the impact of symmetry breaking constraints on the solution of homogeneous instances. We have introduced a new set of heterogeneous instances in order to make comparisons with the homogeneous case. To our knowledge we are the first to solve heterogeneous instances of the MIRP. To this end, we have run a small subset of instances for all three cases with three and four vehicles. We have opted not to change the overall capacity when the vehicles are heterogeneous, but to split it differently among the vehicles. For $K = 3$ the first vehicle accounts for 50% of the original capacity, the second vehicle holds 30% of the original capacity and the third vehicle has 20% of the original capacity. For $K = 4$ the percentage of the original capacity of each vehicle is 40, 25, 20 and 15. Average results are shown in Table 4.7. They indicate that imposing symmetry breaking constraints in the homogeneous MIRP has a significant effect on the reduction of the optimality gap and on computing times. As a result, more instances can be solved optimally. Heterogeneous MIRPs are much easier to solve than homogeneous instances without symmetry breaking constraints. However, they are more difficult than homogeneous instances with these constraints. Note that the CPLEX symmetry breaking reductions parameter was not changed, thus allowing CPLEX to determine which degree of its symmetry breaking should apply.

Table 4.2: Average heuristic solution values for the single vehicle IRP

Periods	Instance	z^*	z	gap (%)	time (s)
$p = 3$	small-5-low	1275.86	1275.86	0.00	15.6
	small-10-low	1910.92	1910.92	0.00	40.4
	small-15-low	2207.76	2207.76	0.00	66.8
	small-20-low	2665.58	2665.58	0.00	89.6
	small-25-low	2987.90	2994.30	0.21	134.0
	small-30-low	3292.93	3296.73	0.11	192.2
	small-35-low	3448.84	3461.41	0.36	249.6
	small-40-low	3703.82	3731.74	0.75	306.4
	small-45-low	3867.48	3899.92	0.83	373.8
	small-50-low	4327.15	4379.75	1.21	502.0
	small-5-high	2199.89	2199.89	0.00	21.4
	small-10-high	4337.97	4337.97	0.00	46.2
	small-15-high	5435.80	5435.80	0.00	74.0
	small-20-high	7225.69	7225.69	0.00	95.0
	small-25-high	8982.07	8988.46	0.07	136.8
	small-30-high	10918.30	10925.18	0.06	189.4
	small-35-high	11411.67	11427.08	0.13	237.0
	small-40-high	12541.05	12577.12	0.28	313.2
	small-45-high	13865.33	13898.74	0.24	362.6
	small-50-high	15410.82	15445.44	0.22	464.8
$p = 6$	small-5-low	3136.90	3136.90	0.00	63.4
	small-10-low	4612.50	4612.50	0.00	143.6
	small-15-low	5418.55	5471.16	0.97	254.0
	small-20-low	6625.35	6672.08	0.70	386.4
	small-25-low	7261.77	7316.62	0.75	513.6
	small-30-low	7710.01	7835.27	1.62	784.2
	small-5-high	5354.20	5354.20	0.00	68.0
	small-10-high	8601.91	8601.91	0.00	143.2
	small-15-high	11543.04	11602.04	0.51	249.2
	small-20-high	14594.13	14714.86	0.82	347.4
	small-25-high	16913.97	17096.76	1.08	535.0
	small-30-high	20410.65	20648.68	1.16	737.2
Average				0.37	

Table 4.3: Average solution values for the small basic MIRP instances, $p = 3$

Instance	$K = 2$							$K = 3$						
	Exact solutions				ALNS			Exact solutions				ALNS		
	# solved	UB	gap (%)	time (s)	UB	gap (%)	time (s)	# solved	UB	gap (%)	time (s)	UB	gap (%)	time (s)
small-5-low	5	1589.87	0.00	3.8	1572.27	0.00	27.2	5	1963.11	0.00	4.6	1963.11	0.00	3480.0
small-10-low	5	2386.95	0.00	7.6	2349.42	0.00	3719.8	5	2899.30	0.00	17.4	2850.57	0.00	3684.2
small-15-low	5	2555.08	0.00	11.8	2536.31	0.00	4207.2	5	2952.27	0.00	31.4	2911.20	0.00	4105.6
small-20-low	5	3075.74	0.00	24.4	3084.59	0.29	3992.4	5	3558.42	0.00	220.8	3563.21	0.13	4769.2
small-25-low	5	3372.11	0.00	31.6	3373.87	0.05	4965.6	5	3860.70	0.00	574.2	3865.72	0.13	4821.6
small-30-low	5	3575.82	0.00	61.8	3603.26	0.77	5587.4	5	3944.38	0.00	1285.8	3985.43	1.04	4687.4
small-35-low	5	3734.30	0.00	56.0	3811.30	2.06	5759.6	5	4214.79	0.00	1935.8	4292.73	1.85	5195.0
small-40-low	5	3982.07	0.00	525.0	4104.21	3.07	6955.2	5	4344.27	0.00	9092.0	4451.24	2.46	5048.6
small-45-low	5	4150.50	0.00	3867.8	4324.40	4.19	6175.0	2	4522.32	0.86	31805.2	4681.85	4.43	5173.4
small-50-low	4	4669.22	0.15	10796.6	4841.90	3.85	6854.8	0	5615.64	12.40	42930.4	5391.42	9.82	4924.2
small-5-high	5	2512.18	0.00	2.8	2494.64	0.00	3032.2	5	2879.54	0.00	2.6	2879.29	0.00	3350.2
small-10-high	5	4811.71	0.00	5.8	4774.99	0.00	3896.2	5	5323.43	0.00	12.6	5276.68	0.00	3540.8
small-15-high	5	5782.28	0.00	11.6	5768.59	0.00	4463.0	5	6182.55	0.00	26.0	6143.17	0.00	4256.4
small-20-high	5	7635.60	0.00	23.8	7644.25	0.11	5213.8	5	8114.74	0.00	217.4	8130.39	0.19	4712.8
small-25-high	5	9369.79	0.00	30.8	9395.88	0.28	5193.4	5	9861.41	0.00	1013.6	9849.23	0.00	5006.4
small-30-high	5	11202.52	0.00	70.4	11230.10	0.25	5926.6	5	11564.18	0.00	1623.4	11590.26	0.23	4978.6
small-35-high	5	11696.90	0.00	65.6	11765.62	0.59	6042.2	5	12175.40	0.00	2696.0	12251.50	0.63	5124.8
small-40-high	5	12827.40	0.00	478.5	12938.10	0.86	6129.0	5	13192.84	0.00	6312.4	13374.34	1.38	5267.8
small-45-high	5	14156.46	0.00	1595.0	14325.60	1.19	5836.4	4	14542.82	0.40	32820.6	14692.62	1.43	5189.2
small-50-high	5	15752.74	0.00	4431.6	15895.42	0.91	6476.2	0	16670.24	4.39	42990.8	16488.44	3.45	5249.0

Table 4.4: Average solution values for the small basic MIRP instances, $p = 6$

Instance	$K = 2$							$K = 3$						
	Exact solutions				ALNS			Exact solutions				ALNS		
	# solved	UB	gap (%)	time (s)	UB	gap (%)	time (s)	# solved	UB	gap (%)	tmie (s)	UB	gap (%)	time (s)
small-5-low	5	3924.29	0.00	9.0	3926.47	0.06	3113.2	5	4991.22	0.00	56.0	4990.03	0.00	2976.4
small-10-low	5	5755.26	0.00	607.4	5793.91	0.67	4069.6	4	6977.63	0.98	14578.6	7177.62	4.08	3158.4
small-15-low	5	6328.64	0.00	555.2	6433.08	1.65	5413.8	4	7378.12	0.36	18761.4	7607.57	3.52	3196.6
small-20-low	5	7608.35	0.00	8642.4	7875.37	3.51	5431.0	1	8993.84	7.11	42751.8	9320.24	12.36	3357.4
small-25-low	4	8175.67	0.24	19002.0	8605.21	5.25	5475.6	0	9625.56	8.76	43047.4	10234.46	17.18	3549.2
small-30-low	1	8443.99	1.74	36841.8	9054.79	7.23	7362.2	0	9723.06	12.52	43079.6	10290.92	21.14	3456.2
small-5-high	5	6140.70	0.00	9.0	6147.72	0.11	3561.2	5	7206.16	0.00	38.0	7206.68	0.01	2851.6
small-10-high	5	9752.96	0.00	57.6	9803.98	0.52	4504.6	4	10975.80	0.71	14611.2	11053.62	1.47	2684.0
small-15-high	5	12457.06	0.00	351.0	12601.52	1.16	4460.0	4	13511.98	3.82	12470.4	13814.68	2.81	2891.8
small-20-high	5	15597.40	0.00	4035.8	15934.08	2.16	5520.8	0	16942.74	3.82	42985.6	17285.32	6.23	3079.4
small-25-high	5	17844.58	0.00	10160.2	18194.68	1.96	6259.4	1	19299.70	4.58	39241.4	19573.78	6.58	3046.0
small-30-high	3	21160.84	0.67	28788.8	21706.46	2.58	5819.4	0	22521.10	6.20	42963.8	22916.90	8.43	2874.0

Table 4.5: Average ALNS solution values for the large basic MIRP instances, $p = 6$

Instance	Number of vehicles			
	$K = 2$	$K = 3$	$K = 4$	$K = 5$
large-50-low	13049.91	14249.57	18450.18	21260.23
large-100-low	25546.13	23591.50	34722.01	37561.98
large-200-low	46524.72	48225.70	63351.94	73145.96
large-50-high	32585.83	33926.45	37972.05	39836.93
large-100-high	60773.11	64562.34	72772.20	75192.23
large-200-high	121982.72	132976.90	141319.30	144866.10

Table 4.6: Average branch-and-cut computational results on the larger instance set, $K = 2$ and 3

Instance	$K = 2$			$K = 3$		
	# best	gap (%)	time (s)	# best	gap (%)	time (s)
large-50-6-high	10	4.00	86400.0	0	15.72	86400.0
large-100-6-high	1	32.57	86400.0	0	56.09	86400.0
large-50-6-low	10	10.94	86400.0	0	36.22	86400.0
large-100-6-low	0	66.49	86400.0	0	77.56	86400.0
Average	5.25	28.50	86400.0	0.00	46.39	86400.0

Table 4.7: Computational results on the MIRP with an homogeneous fleet (with and without symmetry breaking constraints) and with an heterogeneous fleet

K	Instance	Homogeneous, with symmetry breaking			Homogeneous, without symmetry breaking			Heterogeneous		
		# solved	gap (%)	time (s)	# solved	gap (%)	time (s)	# solved	gap (%)	time (s)
3	small-5-3-low	5	0.00	4.6	5	0.00	3.0	5	0.00	2.6
	small-10-3-low	5	0.00	17.4	5	0.00	47.6	5	0.00	14.0
	small-15-3-low	5	0.00	31.4	5	0.00	2218.8	5	0.00	121.6
	small-20-3-low	5	0.00	220.8	2	5.03	13698.4	5	0.00	161.6
	small-25-3-low	5	0.00	574.2	3	1.30	21467.6	4	1.49	15681.2
	small-30-3-low	5	0.00	1285.8	3	4.71	18211.6	2	3.36	19044.2
	small-35-3-low	5	0.00	1935.8	1	5.67	30285.0	4	2.90	18449.8
Average		5	0.00	581.4	3.42	2.38	12276.0	4.28	1.10	7639.28
4	small-5-3-low	5	0.00	4.0	5	0.00	31.0	5	0.00	4.6
	small-10-3-low	5	0.00	40.8	3	1.82	9346.8	5	0.00	25.0
	small-15-3-low	5	0.00	119.0	1	5.48	18586.6	5	0.00	1082.0
	small-20-3-low	5	0.00	5544.4	0	14.12	27856.2	4	2.26	9040.8
	small-25-3-low	5	0.00	4665.8	0	16.32	28606.2	3	4.68	24302.0
	small-30-3-low	2	2.90	29714.8	0	12.61	43200.0	1	8.60	38771.0
	small-35-3-low	2	5.49	31756.2	0	12.11	43200.0	1	5.46	43200.0
Average		4.14	1.19	10263.5	1.28	8.92	24403.8	3.42	3.00	16632.2

4.5.3 Computational experiments for the consistent MIRP

We also report in Tables 4.8 to 4.13 the heuristic solution values of the consistent MIRP for each of the six features described in Section 4.2.2. The last line provides the average percentage increase of each consistent MIRP solution value with respect to the basic MIRP solution values (column $K = 3$ in Tables 4.3−4.5). Specifically, Tables 4.8 to 4.10 report statistics for each set of the low inventory cost instances, starting with three periods and five customers, and going up to six periods and 200 customers, when compared to the solution obtained by our heuristics for the general problem. Tables 4.11 to 4.13 provide statistics for the high inventory cost instances. The parameters we have used to run the tests for each type of consistency are the following:

- Quantity consistency: each delivery performed to any customer must lie within one and three times the average demand of the customer, that is $g_l = 1.0$ and $g_u = 3.0$.
- Vehicle filling rate: each dispatched vehicle must be at least 50% filled, i.e. $\gamma = 0.5$.
- Driver partial consistency: we have tested several different values for the penalty parameter, as reported later; for these tables, we provide results with $\alpha = 10$.
- Visit spacing: a customer may not be visited more than once in every two periods and should be visited at least once in every three periods, i.e. $m_i = 1$ and $M_i = 2$. We did not need to consider customer-dependent values since the instances were generated taking the capacity/demand ratio into account.

The following conclusions can be drawn from Tables 4.8 to 4.13. We have shown that imposing restrictions on the quantities delivered increases the solution cost by at least 1% and by up to 27% in some sets of instances when one forces the delivered quantity to meet customer-dependent intervals, or by as much as 20% when the OU policy is enforced. Simplifying the decision process by applying the OU inventory policy increases the solution cost by more than 9% on average. This finding is consistent with the observation made in Archetti et al. (2007) for the IRP, in Coelho et al. (2012a) for the IRP with transshipment, and in Archetti et al. (2011) for the integrated production-distribution problem.

Imposing a high vehicle capacity utilization rate seems to be the most expensive consistency feature we have tested, especially on instances with many customers. On

Table 4.8: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, low inventory cost, small instances, $p = 3$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
small-5-low	1963.11	2071.22	1990.31	2043.97	2027.53	1967.20	2116.30
small-10-low	2850.57	2872.31	3101.05	3062.11	2913.29	2859.45	3001.69
small-15-low	2911.81	3042.45	3073.49	3184.83	2952.26	2921.35	2986.88
small-20-low	3563.21	3625.43	4131.66	3983.33	3566.44	3552.85	3644.36
small-25-low	3865.72	3940.99	4830.65	4335.33	3877.88	3862.27	3877.81
small-30-low	3985.43	4070.08	4804.55	4339.65	3976.88	3976.12	3995.84
small-35-low	4292.73	4311.82	5377.08	4761.42	4280.70	4321.97	4279.98
small-40-low	4451.24	4556.18	5960.80	4964.44	4454.59	4506.95	4463.59
small-45-low	4681.85	4924.37	7135.81	5217.91	4652.14	4699.30	4641.69
small-50-low	5391.42	5660.46	8238.81	6094.35	5374.27	5429.03	5404.85
Average % increase		2.87	24.23	10.12	0.50	0.31	1.53

Table 4.9: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, low inventory cost, small instances, $p = 6$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
small-5-low	4990.03	5166.03	5048.28	5849.75	5349.59	5023.87	5094.61
small-10-low	7177.62	7294.13	7400.04	7640.90	7263.38	7139.06	7401.88
small-15-low	7607.57	8024.60	8037.87	8120.28	7774.26	7668.43	7721.73
small-20-low	9320.24	9532.51	9740.61	9692.86	9247.78	9301.70	9416.01
small-25-low	10234.46	11141.80	11171.93	10707.82	10142.26	10285.73	10477.21
small-30-low	10290.92	10830.36	11478.12	10986.84	9993.56	10316.08	10649.92
Average % increase		4.54	5.88	6.67	0.93	0.23	0.97

Table 4.10: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, low inventory cost, large instances, $p = 6$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
large-50-low	14249.57	18801.51	21807.19	16884.45	15382.67	14540.93	20178.25
large-100-low	23591.50	32284.46	31505.37	34519.24	29871.05	23048.24	37742.01
large-200-low	48225.70	51122.93	60967.73	57525.15	58337.89	51364.47	47838.81
Average % increase		27.38	39.44	20.70	9.85	3.54	17.48

Table 4.11: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, high inventory cost, small instances, $p = 3$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
small-5-high	2879.29	2989.20	2956.83	2980.34	2946.88	2884.22	3038.88
small-10-high	5276.68	5297.42	5651.45	5510.05	5339.68	5285.29	5429.97
small-15-high	6143.17	6159.05	6619.92	6400.43	6182.55	6152.00	6219.67
small-20-high	8130.39	8170.94	8554.95	8546.77	8121.97	8113.82	8200.63
small-25-high	9849.23	9917.80	10587.54	10319.73	9881.52	9847.79	9878.85
small-30-high	11590.26	11715.32	12536.84	12037.78	11600.30	11621.26	11631.32
small-35-high	12251.50	12336.04	15122.32	12747.70	12211.72	12241.54	12244.00
small-40-high	13374.34	13476.70	16953.84	13800.16	13333.00	13369.60	13335.10
small-45-high	14692.62	15060.74	16784.04	15357.78	14644.72	14731.62	14636.70
small-50-high	16488.44	16872.28	20204.08	17301.32	16482.02	16496.72	16528.04
Average % increase		1.27	12.61	4.22	0.31	0.07	0.96

Table 4.12: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, high inventory cost, small instances, $p = 6$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
small-5-high	7206.68	7383.89	7276.32	8059.51	7551.10	7250.36	7368.82
small-10-high	11053.62	11430.48	11336.22	11546.98	11234.52	11212.76	11399.50
small-15-high	13814.68	13977.72	13951.22	14131.80	13924.24	13761.24	13844.00
small-20-high	17285.32	17536.64	17774.02	17741.98	17274.16	17377.56	17357.76
small-25-high	19573.78	20841.74	20566.58	20312.10	19716.12	19808.78	19720.80
small-30-high	22916.90	23685.94	24109.86	23399.40	22818.68	22972.18	22946.42
Average % increase		3.00	2.89	4.34	1.18	0.57	1.06

Table 4.13: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, high inventory cost, large instances, $p = 6$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
large-50-high	33926.45	37486.37	41449.94	38637.65	35140.25	34470.81	38356.78
large-100-high	64562.34	67306.25	69335.96	73089.95	68054.97	63552.22	67308.71
large-200-high	132076.90	144335.10	134749.30	139010.10	134915.80	129062.10	126762.68
Average % increase		8.07	10.62	10.77	3.75	−0.67	4.51

the other hand, imposing consistency in the assignment of drivers to customers does not change the solution cost if the planning horizon is short, since many customers are served only once. However, allowing some of the deliveries to deviate from the driver consistency rule appears to be a very good feature, since most of the deliveries will still benefit from the driver consistency policy. We believe that the negative average cost increase (an actual cost reduction) shown in Table 4.13 is due to noise, since it only happened on the two largest instance sets and by a small percentage. Adjusting the cost parameter associated with the penalty for assigning more than one vehicle to the same customer can have a major impact both on the consistency of the assignments and on the overall cost. In our tests, the driver consistency and partial consistency policies do not increase solution cost by much.

Ensuring minimum and maximum intervals between successive visits to the same customer usually does not change the solution cost by more than 1.5%, but can be as high as 17% in some cases. Finally, it is also noteworthy that inventory holding costs play a major role not only in the values of the solutions obtained, but also on the performance of the algorithm. From our experiments, the gaps of the different consistency features were larger on the low inventory cost set for all but three cases. This is due to the fact that when inventory costs are low, routing decisions are relatively more important. Generating a good route is significantly harder than obtaining a good inventory replenishment policy, thus the larger gaps when inventory costs are less important.

Based on the findings presented above, we have solved to optimality a subset of the MIRP instances with quantity consistency, the driver consistency and the OU features. Results are summarized in Table 4.14, in which we show the average gap between the best lower and upper bounds and the average running time in seconds. For the sake of comparison we also present the average percentage increase in cost when the upper bounds are compared to those of the basic case without the consistency feature, as presented in Table 4.3. Note that the % increase columns of Table 4.14 are computed over exact solution values, whereas those of Table 4.8 for the same instances were computed for heuristic solution values. However, the results of Table 4.14 are consistent with those of Table 4.8.

Finally, we have conducted experiments to better understand the structure of the solutions in terms of number of routes deployed, number of visits per customer, quantities delivered and average vehicle utilization. Note that these statistics are related to each other. Indeed when vehicle utilization increases, so does the quantity delivered, and fewer visits and routes are needed. In order to quantify these aspects, we have solved the 10 instances containing 15 customers and six periods, allowing

Table 4.14: Average computational results on the small instances set under consistency features

Instance	Quantity consistency			Driver consistency			OU		
	gap (%)	time (s)	% increase	gap (%)	time (s)	% increase	gap (%)	time (s)	% increase
small-5-3-low	0.00	2.8	8.58	0.00	4.0	3.18	0.00	0.6	10.77
small-10-3-low	0.00	12.6	0.35	0.00	26.2	0.48	0.00	14.2	5.33
small-15-3-low	0.00	29.0	0.00	0.00	596.6	0.00	0.00	36.4	6.64
small-20-3-low	0.00	112.6	0.00	3.32	13839.8	0.00	0.00	431.0	9.73
small-25-3-low	0.00	387.4	0.00	3.61	13592.4	0.11	0.00	2752.4	9.59
small-30-3-low	0.00	696.2	0.00	3.12	24191.4	0.09	0.77	10886.2	8.61
small-35-3-low	0.00	1139.0	0.00	0.51	25433.6	0.00	1.76	14799.0	9.53
Average	0.00	339.9	1.27	1.50	11097.7	0.55	0.36	4131.4	8.60

the use of two vehicles under an ML inventory policy. We have then run the same instances under the quantity consistency, the driver consistency and the OU policy consistency features. We provide averages in Table 4.15, presenting the number "# routes" of vehicles dispatched, the number "# visits" of total visits performed, the average size "Avg q" of the deliveries, and the average "Avg $\%Q$" of vehicle utilization. The results indicate that the structure of the solutions, notably the number of dispatched vehicles and quantities delivered do not change considerably when different consistency features are applied. In other words, consistency features seem to affect the cost of the solutions, but not their structure.

Table 4.15: Average computational results on the structure of the solutions with consistency features

Cases	# routes	# visits	Avg q	Avg $\%Q$
Basic	7.0	39.0	98.4	87.6
Quantity consistency	7.0	39.7	96.6	87.6
Driver consistency	7.2	38.9	98.6	85.5
OU policy consistency	7.4	38.8	100.0	83.9

4.5.4 Sensitivity analyses for the consistent MIRP

We have performed some sensitivity analyses on a number of consistency parameters. While some consistency features are hard constraints, like the OU policy and the driver consistency, all others are subject to the influence of parameters that can affect the cost of a solution.

Obviously for the *driver partial consistency* feature, the choice of the value of the parameter α is highly related to the performance of the consistency feature itself and to the cost of the solutions it yields. Thus, we have also evaluated how the *driver partial consistency* case responds to different values of the penalty parameter α. Specifically, we have used $\alpha = 0.1$, 1, 10 and 100. We then observed how many vehicle assignments were made in the final solution, as well as the cost of the solution. As expected, the number of extra vehicles increased in the instances with six time periods, compared with the solutions obtained for the three-period instances. This is due to the fact that many customers were served only once in the shorter horizon instances and automatically respected the driver consistency rule. Also, the number of vehicle assignments decreased to close to one per customer as the value of α increased. Figure 4.3 depicts the average number of vehicle assignments and solution cost per customer.

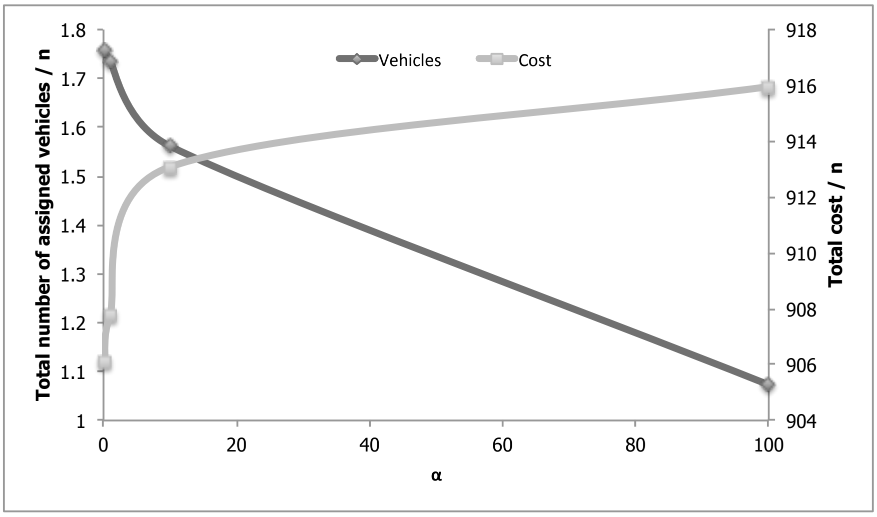

Figure 4.3: Average number of vehicles and cost of the solution per customer for the consistent MIRP with partial driver consistency.

We have also performed sensitivity analyses for the three remaining consistency features. More precisely, we have evaluated how the solution cost is affected by changes in the parameters of the quantity consistency, the vehicle filling rate and the visit spacing features. In particular, for the quantity consistency feature we have performed tests with a loose and with a tight interval; for the vehicle filling rate, we have tested with a lower and a higher vehicle utilization; for the visit spacing, we have run tests with more frequent deliveries and with more spaced deliveries. Results of the experiments on a subset of instances and the values chosen for the parameters are presented in Tables 4.16−4.18. The directions of the cost increases and decreases are small and consistent with the directions of the changes of the parameters.

Table 4.16: Sensitivity analyses for the quantity consistency feature

	$g_l = 0,\, g_u = 4$	$g_l = 1,\, g_u = 2$
Average % cost increase	-1.22	3.11

Table 4.17: Sensitivity analyses for the vehicle filling rate consistency feature

	$\gamma = 30$	$\gamma = 70$
Average % cost increase	-3.15	0.49

4.5.5 Final computational comments

To conclude this section, it is worth making a number of comments on our computational experiments. We have profiled the code of our heuristic algorithm using `GNU gprof` to identify how the computing time was distributed in the algorithm. Our experiments show that approximately 65% of the running time is spent solving network flow problems. This subproblem must be solved several times during the application of some of the ALNS operators, since these need to evaluate the cost of intermediate solutions, and also at the end of each iteration to compute the cost of the new solution. Even though this percentage is high, solving network flow problems is still faster than solving integer linear programs using a general purpose solver. The SI subproblem is solved less often but is more time consuming, taking approximately 7% of the running time. In our experiments we have observed that on average 69% of the calls to SI have led to improvements. Several simple functions that are used very often, usually several times at each iteration, such as identifying the cheapest insertion position, instantiating the DQ and the SI subproblems, removing customers from routes, copying solutions and updating weights and scores, among others, each account for only a few percents of the total running time.

As mentioned in Section 4.4.3 we have opted not to stop the algorithm after a predetermined running time because we wanted to evaluate the relative impact of each policy, and not show how the algorithm performed on any particular one. Thus, even though some computational times are large, our experiments enable us to derive insights on how much each policy would cost to the decision maker, and

Table 4.18: Sensitivity analyses for the visit spacing consistency feature

	$m_i = 0,\, M_i = 3$	$m_i = 2,\, M_i = 2$
Average % cost increase	-0.66	0.89

once he makes his decision, a specific algorithm can be applied to obtain a solution for that particular policy in less time. Specifically, the driver consistency rule yields a high average running time, due to the constraint added to the SI subproblem, with 140,000 seconds on average for the large instances. One particular instance of the driver partial consistency rule ran for almost 30,000 seconds. Simpler models, such as the basic MIRP or the OU policy had an average running time of 2,000 seconds for the small instances with three periods and of 8,000 seconds for the small instances with six periods. On the larger instances, both policies yielded an average of 14,000 seconds.

4.6 Conclusions

We have incorporated six consistency features in the MIRP. One of these is the well-known OU replenishment policy, and another is the concept of driver consistency already introduced in the context of the multi-period VRP. We have extended the branch-and-cut scheme introduced in Chapter 3 to account for multi-vehicles as well as all consistency features. We have also developed a matheuristic composed of an ALNS enhanced by the exact solution of two types of MILPs. The first one is a network flow model used to compute delivery quantities associated with a given set of routes. The second one provides an approximation of the cost of a new solution obtained by applying vertex removals and reinsertions to a given solution. The algorithm is sufficiently flexible to handle the basic MIRP as well as any meaningful combination of the six consistency features we have considered. However, the performance improves when some adjustments are made for certain features. Extensive computational tests on benchmark instances have shown that introducing some of these features can increase the average solution cost significantly, by up to 40% when imposing a high vehicle capacity utilization, or can cost as little as less than 1% when controlling the interval between successive visits to the same customer. Our study clearly illustrates the costs and benefits of incorporating consistency features in the basic MIRP.

Chapter 5

Dynamic and Stochastic Inventory-Routing

Chapter information

An article based on this chapter was submitted for publication in *Transportation Science*: L. C. Coelho, J.-F. Cordeau, G. Laporte. Dynamic and Stochastic Inventory-Routing. Technical Report, *CIRRELT-2012-37*, Montréal, 2012.

In this chapter, we integrate the concepts of *flexibility* and of *consistency* within the framework of inventory-routing. Specifically, we consider a dynamic and stochastic environment, where we compare different policies.

5.1 Introduction

From an operational perspective, the VMI strategy is based on the solution of a difficult combinatorial optimization problem called the Inventory-Routing Problem (IRP), which integrates inventory management and vehicle routing decisions over several periods. The IRP has received increased attention in recent years. Several heuristics (Bertazzi et al., 2002; Archetti et al., 2012; Coelho et al., 2012a) as well as exact algorithms (Archetti et al., 2007; Solyalı and Süral, 2011; Coelho and Laporte, 2013) have been proposed for the single vehicle version of the problem. The multi-vehicle case (MIRP) has also been solved heuristically (Coelho et al., 2012b) and exactly (Adulyasak et al., 2012; Coelho and Laporte, 2013). In addition, an extended version of the MIRP incorporating several consistency features has been solved heuristically and exactly by Coelho et al. (2012b) and Coelho and Laporte

(2013), respectively. However, the studies described in these papers deal with a static and deterministic version of the problem in which all information is available when decisions are made. Literature reviews on the IRP can be found in Campbell et al. (1998), Cordeau et al. (2007) and Andersson et al. (2010).

Dynamic problems are frequently encountered in practice. They reflect real-life situations in which one has to make decisions without full knowledge of future events. Examples of such problems arise in the context of the Dynamic Vehicle Routing Problem in which customer demands are gradually revealed over time (Berbeglia et al., 2010; Wen et al., 2010; Pillac et al., 2011). In Dynamic and Stochastic Inventory-Routing Problems (DSIRP), customer demand is known in a probabilistic sense, thus yielding a dynamic and stochastic problem. In the IRP literature, dynamic problems have been studied by Kleywegt et al. (2002, 2004) who applied dynamic programming, and by Hvattum and Løkketangen (2009) and Hvattum et al. (2009) who used scenario trees and a progressive hedging algorithm. Recently, Bertazzi et al. (2012) have formulated the stochastic IRP as a dynamic program and have solved it by means of a hybrid rolling horizon algorithm. This algorithm estimates unknown demands on the basis of their past average, and then solves a deterministic instance.

Solving a dynamic problem consists of proposing a *solution policy* as opposed to computing a static output (Berbeglia et al., 2010). A possible policy is to optimize a static instance whenever new information becomes available. The drawback of such a method is that it is often very time consuming to solve a large number of instances. A more common policy is to apply the static algorithm only once and then reoptimize the problem through a heuristic whenever new information is made available. A third policy, which can be combined with either of the first two, is to take advantage of the probabilistic knowledge of future information and make use of forecasts. In this chapter we use forecasts in combination with the first policy. For more information on the solution of dynamic problems, see Psaraftis (1998); Ghiani et al. (2003) and Berbeglia et al. (2010).

The deterministic algorithms developed by Coelho et al. (2012a) allow the solution of DSIRPs within a rolling horizon framework, where one uses demand forecasts as an approximation of the future unknown demand. As noted by Özer (2011), the use of past information can become an important aspect of the inventory management process provided it is properly used. Demand forecasts are typically needed for practical inventory control systems, the most common approach being the extrapolation of historical data based on the statistical analysis of time series (Axsäter, 2006).

Our aim is to describe and compare several solution policies for the DSIRP in which the objective is to minimize the total inventory, distribution and shortage costs. There are key differences between our approach and previous ones, in particular that of Bertazzi et al. (2012). One of these lies in the fact that we develop and compare several policies to solve the same problem, instead of only one. In particular, we are able to evaluate the performance of our method on inventory policies that are more general than the (hard constraint) assumption made in that study. Moreover, we propose a method that can make use of historical data in order to take into account future unknown demands, thus being able to efficiently solve instances in which the demand presents a trend or seasonalities, which was not previously the case. We also consider a dynamic environment in which some information arrives over time and is used in a rolling horizon framework. In addition, as in Coelho et al. (2012a), we allow the use of lateral transshipments between customers as a means to avoid stockouts when demand is high. Finally, we evaluate the impact of imposing some consistency features to the solutions of dynamic and stochastic instances of the IRP, thus extending the scope of the study of Coelho et al. (2012b). In addition to proposing an efficient and flexible solution methodology for the DSIRP, one of our main scientific contributions is to evaluate the value of demand forecasts and transshipments.

The remainder of the chapter is organized as follows. In Section 5.2 we formally define the DSIRP and we describe in Section 5.3 the strategies we have developed to solve it. Implementation details are provided in Section 5.4. This is followed by the results of extensive computational experiments in Section 5.5, and by conclusions in Section 5.6.

5.2 Problem description

We now formally introduce the DSIRP. The problem is defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{0, ..., n\}$ is the vertex set and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ is the arc set. Vertex 0 is a depot at which the supplier is located and the vertices of $\mathcal{V}' = \mathcal{V} \setminus \{0\}$ represent customers. The problem is defined over an horizon of length p and at each time period $t \in \mathcal{T} = \{1, ..., p\}$ the demand d_i^t of customer i is a random variable D_i^t. In practice, the demand is not known by the decision maker who has to estimate it on the basis of historical data. We assume the decision maker can use any kind of forecast and input this information into the algorithmic framework we provide. The decision maker realizes the actual values of d_i^t at the end of each period t. A unit inventory holding cost h_i is incurred by customer i and by the supplier at each period, and customer i has an inventory holding capacity C_i. We assume the supplier has enough inventory to meet all the demand during the planning horizon. If the demand of customer i is higher than its inventory level, it is then lost and a unit shortage penalty p_i is incurred. At the beginning of the planning horizon the decision maker knows the inventory level I_0^0 and I_i^0 of the supplier and customer i, respectively.

As is common in the IRP literature, we assume that a single vehicle of capacity Q is available (Bertazzi et al., 2002; Archetti et al., 2007, 2012; Bertazzi et al., 2002, 2012; Coelho et al., 2012a). The vehicle is able to perform one route per time period, from the supplier to a subset of customers. A routing cost c_{ij} is associated with arc $(i, j) \in \mathcal{A}$. We also consider that the supplier uses an order-up-to inventory policy. This policy has been widely used in IRPs and related problems (Bertazzi et al., 2002; Archetti et al., 2007, 2012; Adulyasak et al., 2012; Coelho et al., 2012a) and ensures that whenever a customer is visited, the quantity delivered is that needed to fill its inventory capacity. To ensure the feasibility of such a policy, given that there is only one capacitated vehicle available, we assume direct deliveries can take place from the supplier to any customer, by subcontracting to a carrier, to allow for planned deliveries to meet the OU requirements. In addition, after the demand is realized, if a customer faces a shortage it can arrange a lateral emergency transshipment from another customer if this is feasible. Both types of outsourced deliveries (direct deliveries and lateral emergency transshipments) are only made by direct shipping and the unit cost associated with direct deliveries or transshipments from i to j is βc_{ij}, where $\beta > 0$. As is standard in vehicle routing, travel costs are distance-dependent and are unrelated to the vehicle load. However, direct delivery and transshipment costs are distance- and volume-dependent because this is often how outsourced carriers define the terms of their contracts.

Regarding temporal issues, we consider that the decision maker first decides which customers to replenish in each period as well as the associated vehicle route and the direct shipments, if any. After demand is revealed, lateral transshipments may be arranged if any customer faces a shortage.

The variables and constraints of the model are as follows. Let I_i^t be the inventory level at customer i at the end of period t, q_i^t the quantity delivered to customer i in period t using the supplier's vehicle, w_{ij}^t the quantity carried by the outsourced carrier from customer i to customer j in period t, and l_i^t the lost demand at customer i in period t due to insufficient inventory. The inventory level at the end of period t at customer i is then

$$I_i^t = I_i^{t-1} + q_i^t + \sum_{j \in \mathcal{V}} w_{ji}^t - \sum_{j \in \mathcal{V}'} w_{ij}^t - d_i^t + l_i^t \qquad i \in \mathcal{V}' \quad t \in \mathcal{T}'. \tag{5.1}$$

The objective is to minimize the total inventory, shortage, routing an transshipment costs over the planning horizon, that is

$$\text{minimize} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{V}} h_i I_i^t + \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{V}'} p_i l_i^t + \beta c_{ij} \sum_{t \in \mathcal{T}} \sum_{i,j \in \mathcal{V}'} w_{ij}^t + c_{r_t}, \tag{5.2}$$

where c_{r_t} represents the cost of the route performed in period $t \in \mathcal{T}$, which can be obtained by solving a Traveling Salesman Problem over all the customers visited in period t.

5.3 Solution policies

The problem can be solved under a *proactive* policy or under a *reactive* policy, depending on whether demand forecasts are made or not. For each of these two policies emergency lateral transshipments can be allowed or not. This yields a total of four policies, which are all implemented in a rolling horizon fashion.

5.3.1 Reactive policies

Under reactive policies, which are sometimes called “wait and see”, one observes the state of the system in order to make the next decision regarding routing and delivery. Formally, a reactive policy is defined as an (s, S) replenishment system under which whenever the inventory reaches the reorder point s, it triggers a replenishment order to bring the inventory position up to level S. The reorder point s should consider the delivery lead time and the stockout risk resulting from the stochasticity of the demand.

5.3.1.1 Routing only

Under this policy, deliveries are performed by the supplier’s vehicle and no emergency lateral transshipment takes place when a customer runs out of inventory. Routing decisions are based solely on a customer-dependent threshold s_i and on its inventory level. If the inventory level at customer i is below s_i when the actual demand is realized at the end of period t, then customer i is selected to be served in period $t + 1$. The threshold can be updated after each period. The replenishment level S_i usually depends on ordering and holding costs and is set to bring the inventory level up to a target value. This inventory policy has been widely studied and used in other IRP studies (Bertazzi et al., 2002, 2012; Archetti et al., 2012; Coelho et al., 2012b,a). As noted by Bertazzi et al. (2012) and Coelho et al. (2012b), the OU policy is also relevant from a practical point of view and simplifies the decision making process while ensuring the stability and consistency of the replenishments. As in these studies, we also assume that the target level meets the customer inventory capacity. As mentioned, in order to ensure that this rule is always met and to avoid infeasibilities due to insufficient vehicle capacity, direct deliveries are allowed to take place from the depot. This ensures that all customers i whose inventory level is below the threshold s_i will have their inventories filled to their capacity in the next period.

5.3.1.2 Routing and transshipment

This policy allows lateral transshipments between customers as an emergency measure against stockouts. The decision regarding whether or not to visit customer i is dependent on the threshold s_i as before. The inventory policy applied still follows an OU policy in which direct deliveries are allowed to take place from the supplier. After these decisions have been made, demand is revealed. If a customer runs out of inventory when its demand is realized, lateral transshipments can take place whenever they are possible and economically interesting. Lateral transshipments are allowed only as an emergency measure, i.e. they cannot be used to move inventory to a location having a lower holding cost. This policy is in line with the description of emergency transshipments provided by Nonås and Jörnsten (2007) and Paterson et al. (2011).

5.3.2 Proactive policies

A proactive policy not only observes the state of the system but also attempts to anticipate its future state by forecasting the demand and by using this information in the planning process.

5.3.2.1 Routing only

This policy makes use of forecasts as a means of taking into account future demand but does not allow lateral transshipments. Once forecasts are obtained, the problem can be solved as a deterministic IRP. Direct deliveries from the supplier to the customers are allowed to ensure the feasibility of the OU policy. Under this policy, we first compute an f-period forecast for each of the customers, on the basis of their historical demands. A prediction interval that makes use of probabilistic information is computed for each customer. Forecasts are then used as a proxy for the unknown demands and initial inventory levels are set equal to the last known inventory level of each customer. The problem can then be solved heuristically as a deterministic IRP. The algorithm provides an f-period plan, of which only the first-period solution is implemented. Demands are then realized, new forecasts are computed and the process is reiterated.

5.3.2.2 Routing and transshipment

As an extension of the previous policy, in this case lateral transshipments are allowed to take place after the demand is realized.

5.4 Algorithms

In this section we describe the four algorithms resulting from the solution policies described in Section 5.3.

5.4.1 Reactive policies

We first describe the two algorithms proposed to implement the reactive policies, with or without the use of lateral transshipments.

5.4.1.1 Routing only

The first decision made under this policy regards the level of the inventory at which the reorder point s_i of customer i is set. It is equal to an estimate of the expected demand during the lead time L, plus a safety stock dependent on demand variability, lead time and target service level. We denote the estimate of the expected demand μ_i of customer i per period by $\hat{\mu_i}$ and that of its standard deviation σ_i by $\hat{\sigma_i}$. These values as well as the resulting threshold can be updated at every period. Following classical inventory management practices (Eppen and Martin, 1988), and assuming independent and normally distributed demands, s_i can be computed as

$$s_i = L\hat{\mu_i} + z_\alpha \sqrt{\hat{\sigma_i}^2 L}, \tag{5.3}$$

where α is the probability of a stockout and z_α is the α-order quantile of the demand distribution. The quantity $1 - \alpha$ is usually referred to as the *service level.*

The selection of customers to serve with the supplier's vehicles and through direct deliveries, as well as the quantities delivered by each option yields an NP-hard problem. However, since these decisions should be taken only once for every period, and given the size of the instances considered in this study, we have decided to solve this problem exactly by means of a mixed-integer linear program (MILP). The problem is defined as follows.

If the inventory level of customer i is below its threshold s_i, the total quantity that must be delivered is then the one needed to fill its capacity (i.e. an OU policy applies); otherwise no delivery is made. This quantity defines the parameter d'_i. We then solve the following MILP, called Routing-Direct (RD), in order to decide which customers are visited by the supplier's vehicle, which ones are visited through direct deliveries (and combinations of these two options), and the quantities delivered by each mode. When the routing cost matrix c_{ij} is symmetric, as is the case in our computational experiments, we work with an undirected formulation in order to reduce the number of variables. Thus, the routing variables $x_{ij} (i < j)$ are equal to the number of times edge (i, j) is traversed. We also introduce binary variables y_i, equal to one if and only if vertex i (the supplier or a customer) is visited by the supplier's vehicle. We denote by q_i the quantity delivered by the supplier's vehicle and by w_i the quantity delivered through direct deliveries to customer i. The problem can then be formulated as follows:

$$\text{(RD) minimize} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}, i<j} c_{ij} x_{ij} + \beta \sum_{i \in \mathcal{V}} w_i c_{0i}, \tag{5.4}$$

subject to

$$q_i + w_i = d'_i \quad i \in \mathcal{V}' \tag{5.5}$$

$$\sum_{i \in \mathcal{V}'} q_i \leq Q \tag{5.6}$$

$$q_i \leq Q y_i \quad i \in \mathcal{V}' \tag{5.7}$$

$$\sum_{j \in \mathcal{V}, i<j} x_{ij} + \sum_{j \in \mathcal{V}, j<i} x_{ji} = 2y_i \quad i \in \mathcal{V} \tag{5.8}$$

$$\sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}, i<j} x_{ij} \leq \sum_{i \in \mathcal{S}} y_i - y_m \quad \mathcal{S} \subseteq \mathcal{V}', \text{ for some } m \in \mathcal{S} \tag{5.9}$$

$$q_i, w_i \geq 0 \quad i \in \mathcal{V}' \tag{5.10}$$

$$x_{i0} \in \{0, 1, 2\} \quad i \in \mathcal{V}' \tag{5.11}$$

$$x_{ij} \in \{0, 1\} \quad i, j \in \mathcal{V}' \tag{5.12}$$

$$y_i \in \{0, 1\} \quad i \in \mathcal{V}. \tag{5.13}$$

The objective function (5.4) defines the minimization of routing and direct delivery costs. Constraints (5.5) state that the total delivered quantity d'_i is equal to the quantity q_i delivered by the supplier's vehicle, plus the quantity w_i supplied by means of a direct delivery. Constraints (5.6) ensure that the vehicle capacity is not exceeded, while constraints (5.7) guarantee that only customers assigned a visit can have quantities delivered by the supplier vehicle. Constraints (5.8) and (5.9) are degree constraints and subtour elimination constraints, respectively. Constraints (5.10)−(5.13) enforce the integrality and non-negativity conditions on the variables.

The RD model can be simplified by preprocessing all customers with zero d'_i and removing the corresponding variables.

Algorithm 5.1 provides a pseudocode of this policy implementation.

Algorithm 5.1 Pseudocode: Routing only

```
for $t = 0$ to $p - 1$ do
    for $i = 1$ to $n$ do
        Compute $s_i$ on the basis of the past demand.
        if $I_i^t < s_i$ then
            Include $i$ in the set of customers that follow an OU policy in period $t + 1$.
        end if
    end for
    Solve RD to define direct deliveries destinations and quantities.
end for
```

5.4.1.2 Routing and transshipment

The implementation of this policy is like the previous one except that after the solution has been computed, demands are revealed and lateral transshipments are allowed to take place. These are computed by means of the following min-cost network flow problem. This model, called Transshipment Origins-Destinations (TOD), optimizes the quantities as well as origins and destinations of the lateral transshipments. Note that in this model the parameter I_i^0 represents the initial inventory of customer i at the beginning of each time slice of the rolling horizon, unlike the initial inventory of the instance being solved as it was defined in Section 5.2. It is solved once per period, after demands are realized. The problem is defined as follows:

$$\text{(TOD)} \quad \text{minimize} \quad \beta \sum_{i \in \mathcal{V}'} \sum_{j \in \mathcal{V}'} c_{ij} w_{ij} + \sum_{i \in \mathcal{V}'} p_i l_i + \sum_{i \in \mathcal{V}'} I_i h_i \tag{5.14}$$

subject to

$$I_i = I_i^0 + \sum_{j \in \mathcal{V}'} w_{ji} - \sum_{j \in \mathcal{V}'} w_{ij} + l_i \quad i \in \mathcal{V}' \tag{5.15}$$

$$0 \leq I_i \leq C_i \qquad i \in \mathcal{V}' \tag{5.16}$$

$$0 \leq l_i \leq -\min\{0, I_i^0\} \quad i \in \mathcal{V}' \tag{5.17}$$

$$0 \leq w_{ij} \leq \min\{\max\{0, I_i^0\}, -\min\{0, I_j^0\}\} \quad i, j \in \mathcal{V}'. \tag{5.18}$$

The objective function (5.14) minimizes the total lateral transshipment, lost demand and inventory costs. Constraints (5.15) ensure flow conservation by stating that the final inventory of customer i is the sum of its initial inventory, plus all quantities transshiped to i, minus all quantities transshiped from i to other customers, plus the lost demand. Constraints (5.16) set bounds on the final inventory. Constraints (5.17) define bounds on the lost demand of customer i: if its initial inventory is non-negative, then no demand can be lost, and both bounds are zero; otherwise, a minimum of zero and a maximum of I_i^0 units can be lost. Likewise, constraints (5.18) impose bounds on the flows of transshipment arcs. There are four possible combinations of inventory levels for i and j, all of which can be handled by these constraints:

1. $I_i^0 \geq 0$ and $I_j^0 \geq 0$: the inner $\min\{0, I_j^0\}$ is zero, setting the right-hand side of the constraint to zero. No transshipment should occur only to relocate inventory, since j does not need an emergency transshipment;

2. $I_i^0 \geq 0$ and $I_j^0 < 0$: the inner $\min\{0, I_j^0\}$ is I_j^0 since this quantity is negative; the outer $\min\{I_i^0, I_j^0\}$ is then the minimum between the availability I_i^0 and the requirement $-I_j^0$. This is then the upper bound on the arc of the emergency transshipment from i to j;

3. $I_i^0 < 0$ and $I_j^0 \geq 0$: both inner functions return zero; the upper bound is then also zero, since j does not need an emergency transshipment and i does not have a surplus;

4. $I_i^0 < 0$ and $I_j^0 < 0$: the max function returns zero and the inner min function returns $-I_j^0$; the outer function then becomes $\min\{0, I_j^0\}$ which returns zero as the upper bound flow on the arc flow, since i does not have enough inventory to supply to j.

We depict in Figure 5.1 a simple example of this network flow problem. The flow on the small dashed arcs equals the initial inventory level at customer i. Note that this number represents the surplus available at vertex i. If I_i^0 is negative, it will enable customer i to have a lost demand. Then the flow over the large dashed arcs lies in the interval $[0, -\min\{0, I_i^0\}]$ and represents the lost demand of customer

i. Note that if I_i^0 is positive, the flow on this arc is set to zero; if I_i^0 is negative, it represents the lost demand and lies between zero and $-I_i^0$, i.e. this is the case in which all the excess demand is lost. The costs of these arcs are equal to p_i. The solid arcs represent the inventory carried at customer i at the end of the period. The flows on these arcs are bounded by $[0, C_i]$ and their associated costs are h_i. Finally, the dotted arcs in the middle represent transshipments. They are defined between any pair of vertices (i, j), in both directions, and their cost is βc_{ij}. The flows on these arcs lie in the interval $\Big[0, \min\{\max\{0, I_i^0\}, -\min\{0, I_j^0\}\}\Big]$.

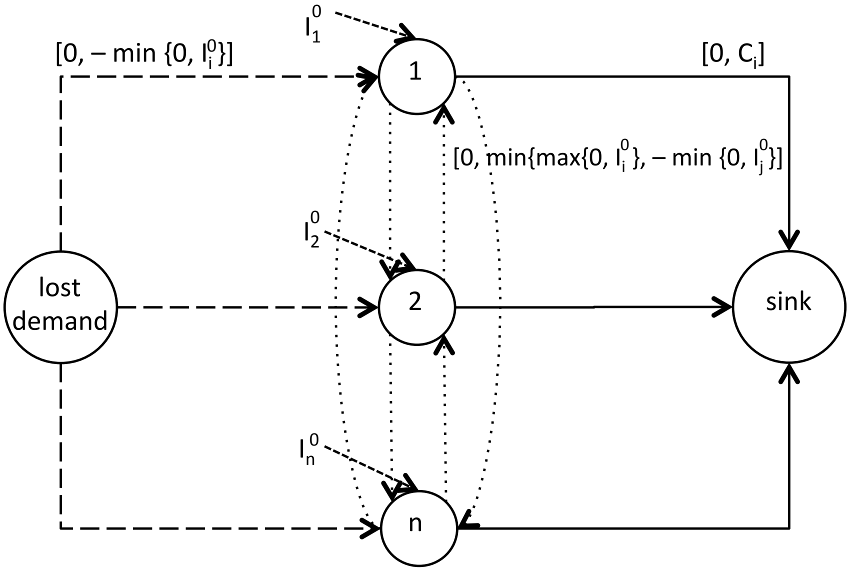

Figure 5.1: Example of the network flow problem solved to decide of transshipment quantities, origins and destinations.

The pseudocode corresponding to this policy is described in Algorithm 5.2.

5.4.2 Proactive policies

We now describe the two algorithms used to implement the proactive policies.

5.4.2.1 Routing only

This policy makes use of forecasts on future demand to help make current decisions. The first decision relates to the choice of a forecasting method. There exist several methods for forecasting future demand based on time series analysis. For an overview, see Makridakis et al. (1998). In this chapter we apply the exponential smoothing technique which assigns exponentially smaller weights to past observations. This is a simple yet powerful method capable of identifying changes in the mean, trend or seasonalities in time series. It provides a point forecast, i.e. a single

Algorithm 5.2 Pseudocode: Routing and transshipment

```
for $t = 0$ to $p - 1$ do
  for $i = 1$ to $n$ do
    Compute $s_i$ on the basis of the past demand.
    if $I_i^t < s_i$ then
      Include $i$ in the set of customers that follow an OU policy in period $t + 1$.
    end if
  end for
  Solve RD to define direct deliveries destinations and quantities.
  Reveal demands of period $t + 1$.
  for $i = 1$ to $n$ do
    if $I_i^t + q_i^{t+1} - d_i^{t+1} < 0$ then
      Allow transshipments to customer $i$.
    end if
  end for
  Solve TOD to define transshipments origins/destinations and quantities.
end for
```

value representing the expected future demand, or a prediction interval, i.e. a point forecast and an estimated variance (see Hyndman et al. (2008)).

The second decision regards the length f of the forecasting and rolling horizon. A compromise must be made between a short horizon which yields faster computations but lower solution quality, and a longer horizon which considers more information but requires more extensive computations. In Section 5.5.3.5 we examine the impact of f on the solution process.

Finally, the third decision is how to incorporate future demand forecasts in an IRP heuristic. We have adapted the work of Coelho et al. (2012a) which uses an adaptive large neighborhood search (ALNS) matheuristic and provides very good results on benchmark static instances. This heuristic is described in Section 5.4.3 and can handle both the OU policy or the more general maximum level (ML) inventory policy, which does not force the deliveries to fill the customer capacity. Once forecasts are available, the dynamic problem reduces to a static one.

Algorithm 5.3 provides the pseudocode of this policy.

5.4.2.2 Routing and transshipments

This policy works much like the previous one, except that after vehicle routes are created for all periods of the rolling horizon and the first of them is implemented, de-

Algorithm 5.3 Pseudocode: Routing only

```
for $t = 0$ to $p - 1$ do
    for $i = 1$ to $n$ do
        Compute an $f$-period forecast model for $i$ based on past demand observations.
    end for
    Apply the ALNS-based heuristic to the reduced $f$-period problem.
    Implement the route obtained for the first period.
end for
```

mands are revealed and lateral transshipments are allowed as an emergency measure against shortages. The way these transshipments are computed follows the same min-cost network flow problem, as in Section 5.4.1.2. The pseudocode of this policy is presented in Algorithm 5.4.

Algorithm 5.4 Pseudocode: Routing and transshipments

```
for $t = 0$ to $p - 1$ do
    for $i = 1$ to $n$ do
        Compute an $f$-period forecast model for $i$ based on past demand observations.
    end for
    Apply the ALNS-based heuristic to the reduced $f$-period problem.
    Implement the route obtained for the first period.
    Reveal demands of period $t + 1$.
    for $i = 1$ to $n$ do
        if $I_i^t + q_i^{t+1} - d_i^{t+1} < 0$ then
            Allow transshipments to customer $i$.
        end if
    end for
    Solve TOD to define the transshipments' origins, destinations and quantities.
end for
```

5.4.3 ALNS matheuristic

The algorithm proposed by Coelho et al. (2012a) is an implementation of the ALNS algorithm originally proposed by Ropke and Pisinger (2006a) for the Vehicle Routing Problem and already successfully applied to a number of other contexts (Pepin et al., 2009; Bartodziej et al., 2009; Hewitt et al., 2010; Laporte et al., 2010).

In this implementation, some subproblems are solved exactly as min-cost network flow problems. It can therefore be described as a *matheuristic* (Maniezzo et al., 2009), i.e. as a hybridization of a heuristic and of a mathematical programming algorithm. This algorithm is highly suitable for the problem at hand because of its generality and flexibility. It provides a highly diversified search through the multiplicity of its operators and through the use of a random mechanism for their selection. This implementation uses a subset of the operators used in Coelho et al. (2012a) and runs for fewer iterations in order to make it faster. Because of the dynamic nature of our problem, we must indeed be able to run it several times for a single instance. The impact of this implementation choice is analyzed in Section 5.5.3.

In summary, the algorithm of Coelho et al. (2012a) creates different vehicles routes at each ALNS iteration by removing and reinserting customers into vehicle routes. This is done by selecting one of several simple operators to explore different neighborhoods of the incumbent solution. Such operators include random insertions or removals, best insertions or removals, cluster insertions or removals, emptying routes, swapping routes and moving customers assignments. After vehicle routes have been created, the remaining problem is that of determining delivery quantities and transshipment origins, destinations and quantities, while minimizing the total inventory-distribution cost. This problem is solvable efficiently and exactly using a min-cost network flow algorithm and can easily handle both the ML and OU policies. This approach was shown in Coelho et al. (2012a) to generate IRP solutions with value lying within 0.50% of optimality.

Each operator i is assigned a weight ω_i whose value depends on its past performance and on its score. Given h operators with weights ω_i, operator j will be selected with probability $\omega_j / \sum\limits_{i=1}^{h} \omega_i$. Initially, all weights are equal to one and all scores are equal to zero. Operators are rewarded according to the their past performance: they receive a high reward σ_1 if they yield a new best solution, a medium reward σ_2 if their solution is better than the incumbent one, or a low reward σ_3 if the solution they provide is worse but still accepted. Initially, all operators have the same probability of being selected. After φ iterations, scores are computed taking into account the rewards accumulated as follows. Let π_i and o_{ij} be, respectively, the score of operator i and the number of times it has been used in the last segment j. The updated weights are then

$$\omega_i := \begin{cases} \omega_i & \text{if } o_{ij} = 0 \\ (1-\eta)\omega_i + \eta\pi_i / o_{ij} & \text{if } o_{ij} \neq 0, \end{cases} \tag{5.19}$$

where $\eta \in [0, 1]$ is called the reaction factor, controlling how quickly the weight adjustment reacts to changes in the movement performance. All scores are reset to zero.

New solutions are accepted or rejected according to a simulated annealing criterion: given a solution s, a neighbor solution s' is accepted if $z(s') < z(s)$, and with probability $e^{-(z(s')-z(s))/\tau}$ otherwise, where $z(s)$ is the solution cost and $\tau > 0$ is the current temperature. The temperature is initialized at τ_{start} and is decreased by a cooling rate factor ϕ at each iteration, where $0 < \phi < 1$.

We have used the following destroy and repair operators. In what follows, all insertions are performed following the cheapest insertion rule and ρ is an integer randomly drawn from the interval [1, n] using a semi-triangular distribution with a negative slope.

- Destroy operators
  - **Randomly remove $\boldsymbol{\rho}$**: This operator randomly selects one period and removes one randomly selected customer from it. It is repeated ρ times.
  - **Shaw removal**: Following the ideas developed by Ropke and Pisinger (2006a) and Shaw (1997), this operator removes customers that are relatively close to each other. Specifically, it randomly selects one period and one customer served in this period, it computes the distance $dist_{min}$ to the closest customer also being served by the same route, and it removes all customers within $2dist_{min}$ units from the selected route.
  - **Empty one period**: This operator selects one random period and removes all customers assigned to it.
  - **Remove one customer**: This operator randomly selects one customer and removes all its assignments to any periods.
- Repair operators
  - **Randomly insert $\boldsymbol{\rho}$**: This operator randomly inserts ρ customers into the current solution. Specifically, it selects one random customer and one random period, and inserts it into the route in that period if it is not already present. This operator is applied ρ times.
  - **Shaw insertions**: This operator is similar to the Shaw removal operator in the sense that it selects similar customers to be inserted together. It selects one period and one customer not served in that period. The operator then computes $dist_{min}$ and all customers within a $2dist_{min}$ distance are inserted in the same route.

- **Swap ρ customers**: This operator selects two customers from two different periods and swaps their assignments. It is also applied ρ times.
- **Insert one customer several periods**: This operator selects one customer and randomly assigns it to several periods of the planning horizon.

The operators just described generate the selection of visited customers as well as their sequence in the vehicle route. The remaining problem is that of determining delivery quantities and transshipment origins, destinations and quantities, which can be solved very efficiently by means of a min-cost network flow algorithm.

Given that the ALNS algorithm is invoked several times in a rolling horizon fashion, it had to be tuned to be extremely streamlined and fast. This drove us to the following settings for the operators and parameters after a tuning phase. The starting temperature τ_{start} is set to 20,000 and the cooling rate ϕ is 0.9993. The stopping criterion is satisfied when the temperature reaches 0.01, that is, when approximately 20,000 iterations have been performed. In our implementation, the segment length φ was set to 200 iterations and the reaction factor η was set to 0.7, that is, new weights will be composed by 70% of the performance on the last segment and 30% by the last weight value. Scores are updated with $\sigma_1 = 10$, $\sigma_2 = 5$ and $\sigma_3 = 2$.

At the end of each segment we also perform a 2-opt periodic postoptimization. Algorithm 5.5 presents a simplified pseudocode for this heuristic. For algorithmic and implementation details, the reader is referred to Coelho et al. (2012a).

5.5 Computational experiments

In this section we provide some implementation specifications, we describe the generation procedure for the test instances and we present results of extensive computational experiments. These are described in Section 5.5.3.1 for the base case and in Sections 5.5.3.2 to 5.5.3.7 for several alternative configurations.

5.5.1 Implementation specifications

All computations were performed on a grid with 630 nodes available and running the Scientific Linux 6.1 operating system. Each vertex is equipped with two Intel Westmere-EP X5650 hexa-core processors running at 2.67 GHz and with 24 GB or 48 GB of RAM memory.

Our algorithm was coded in C++ and makes use of only one processor. The min-cost network flow problem was implemented using the *LEMON* graph template

Algorithm 5.5 ALNS heuristic - simplified pseudocode

```
Initialize: set all weights equal to 1 and all scores equal to 0.
$s_{best} \leftarrow s \leftarrow$ initial solution.
$\tau \leftarrow \tau_{start}$.
while $\tau > 0.01$ do
    $s' \leftarrow s$.
    Select a destroy and a repair operator and apply them to $s'$.
    Fix routing decisions, solve the remaining network flow problem.
    if $z(s') < z(s)$ then
        $s \leftarrow s'$;
        if $z(s) < z(s_{best})$ then
            $s_{best} \leftarrow s$;
        else if $s'$ is accepted by the simulated annealing criterion then
            $s \leftarrow s'$;
        end if
    end if
    if the iteration count is a multiple of $\varphi$ then
        update the weights and reset the scores of all operators.
        perform an intra-route 2-opt.
    end if
    $\tau \leftarrow \phi\tau$;
end while
return $s_{best}$;
```

library (Dezső et al., 2011) running the network simplex algorithm for its internal computations. Forecasts were carried out using the *forecast* package (Hyndman and Khandakar, 2008; Hyndman et al., 2012) available for *R* Language and Environment for Statistical Computing (R Development Core Team, 2011) and embeded within our C++ code using the *RInside* classes (Eddelbuettel and François, 2012). We allowed the software to run in its default settings, searching through all the 30 variants of the exponential smoothing models described in Hyndman et al. (2008). We made use of the 50 past periods immediately before the current period as historical data for the chosen forecasting method.

Given that the lead time is equal to one (all deliveries are performed in the next period) and its standard deviation is zero, the value of s_i used in equation (5.3) is then

$$s_i = \hat{\mu}_i + z_\alpha \hat{\sigma}_i. \tag{5.20}$$

Using the last known demand as an expectation of future demand is equivalent to a naïve forecast method in which the next period forecast is equal to the last known value, this being the simplest adaptive forecasting method (Goetschalckx, 2011).

5.5.2 Instance generation

We have generated instances following some of the standards used for the instances generated for the IRP by Archetti et al. (2007, 2011), namely the mean customer demand, initial inventories, vehicle capacity and geographical location of the vertices are the same as in their tests. Instances were generated with 50 past periods of demand information before the future p periods such that it can used as historical data. Our set is generated according to the following data:

- number of customers n: $5k$ where $k = 1, 2, 3, 5, 10, 15, 20, 25, 30, 40$;
- horizon p: equal to 5, 10 or 20 periods;
- demand distributions: mean demand μ_i is generated as an integer random number following a discrete uniform distribution in the interval [10, 100], and standard deviation σ_i as an integer random number following a discrete uniform distribution in the interval [2, 10]. The demands are generated following a normal distribution with these parameters. If a negative demand value is generated, it is substituted by zero;
- product availability at the supplier: mean production $\bar{r}$ is generated as an integer random number following a discrete uniform distribution in the interval

[100n, 140n], and σ_0 as an integer random number following a discrete uniform distribution in the interval [2, 10]. The production is generated following a normal distribution with these parameters. They are used only to account for inventory costs at the supplier, as in Archetti et al. (2007);

- maximum inventory level C_i: $\mu_i g_i$, where g_i is randomly selected from the set $\{2, 3, 4\}$;
- starting inventory level I_0^0: $\sum\limits_{i \in \mathcal{V}'} C_i$;
- starting inventory level I_i^0: $C_i - \mu_i$;
- inventory holding cost h_0: 0.01;
- inventory holding cost h_i ($i > 0$): randomly generated from a continuous uniform distribution in the interval [0.02, 0.10];
- shortage penalty: $p_i = 200h_i$;
- vehicle capacity Q: $\frac{3}{2} \sum\limits_{i \in \mathcal{V}'} \mu_i$;
- distance/cost c_{ij}: $\lfloor \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} + 0.5 \rfloor$, where the points (X_i, Y_i) are the coordinates of vertex i and are obtained randomly from a discrete uniform distribution in the interval [0, 500].

This set of instances will be called the *stationary* data set since the mean of the demand distribution is stationary. We have also generated other sets of instances in order to evaluate the dynamics of real-life demand, which often presents some seasonality. Indeed, Bhatnagar and Teo (2009) show that the key challenges faced in practical supply chains are related to, among others, non-stationary demand and inventory imbalances. To this end, we have generated the following two extra sets of instances, called *seasonal* and *correlated*.

In the *seasonal* data set each customer presents an independent seasonal pattern every five periods. This simulates the weekly variations of orders that are likely to occur. In its lower state, the demand is allowed to be as low as 40% of the usual demand, and as high as 200% at its peak. Seasonalities are independent so that, on average, they should cancel each other and the supplier should not face an overall high or low demand on any given day. In the *correlated* data set, on the other hand, all customers present the same seasonality pattern, that is, all have their lower and higher demands in the same period. This way the supplier faces a bottleneck of its vehicle capacity when the demand is high and has spare capacity when the demand is low. All else is kept unchanged from the *standard* stationary data set. We should mention that computing the reorder point with equation (5.20) assumes that demands of consecutive periods are independent, which is no longer the case in the presence of seasonality. However, we believe that computing the reorder point in this approximate way does not have a major impact on the results.

For each of the three data sets, and each of the 30 combinations of n and p, we have generated five instances, yielding 150 instances in each set, for a total of 450 instances. Their nomenclature follows the rule *dirp-n-p-1* through *dirp-n-p-5*. In Section 5.5.3 we provide summaries aggregating instances by their size: those with less than 50 customers are labeled *small*, those containing between 50 and 100 customers are called *medium*, and those with more than 100 customers are called *large* instances. These sets of instances are available at the URL `http://www.leandro-coelho.com/instances/`. For full results on all instances the reader is referred to Appendix A.4.

5.5.3 Computational results

We now report the results of our extensive computational experiments. The OU policy is first used to allow fair comparisons; the ML policy will be analyzed later. The transshipment cost β was set to 0.01 as in Coelho et al. (2012a) and 95% prediction intervals were used, as in Goodwin et al. (2010). We first present results for the base case, and later we provide analyzes for a number of variations of the problem and of the algorithm.

5.5.3.1 Results for the base case

We first provide results for the *base case* defined with the standard data set in Table 5.1 for the cases without and with lateral transshipments. For each method we present the solution cost, the average running time and the average lost demand per customer per period. Regarding the use of forecasts, one should consider that the value added by forecasting a stationary time series is no better than using the naïve method employed by the reactive policy (Darrat and Zhong, 2000). Nevertheless, some conclusions can be drawn from Table 5.1. On average the solution cost is lower and there is significantly less lost demand. More interestingly, allowing transshipments has a twofold effect: first this helps satisfy the demand by decreasing the average lost demand under both the reactive and proactive policies; second, by decreasing the lost demand, it also lowers the average solution cost. Finally, the computational cost of forecasting and solving the ALNS heuristic for many customers and several periods is not negligible.

Table 5.1: Summary of computational results for the Dynamic and Stochastic Inventory-Routing Problem on the standard data set

Transshipment	Instance	Reactive policy			Proactive policy		
	size	Solution	Time (s)	Avg. lost	Solution	Time (s)	Avg. lost
No	small ($n < 50$)	14974.17	0.0	0.62	14224.84	47.2	0.10
	medium ($50 \leq n \leq 100$)	39546.01	4.3	0.41	32774.85	453.3	0.00
	large ($n > 100$)	64854.75	408.5	0.46	64784.85	3781.0	0.00
Average		39791.64	137.6	0.50	37261.51	1427.2	0.03
Yes	small ($n < 50$)	14382.67	0.0	0.00	8586.53	46.9	0.05
	medium ($50 \leq n \leq 100$)	37720.58	4.4	0.00	27743.95	452.7	0.00
	large ($n > 100$)	61455.93	498.4	0.00	56506.38	3934.4	0.00
Average		37853.06	167.6	0.00	30945.62	1478.0	0.02

In Table 5.2 we provide results for the base case defined with the seasonal data set. Our first observation is that the average running time is higher than in the standard data set. This reflects the difficulty of solving these instances. The value of lateral transshipments is corroborated, as in the standard case: allowing transshipments reduces the average lost demand per customer per period while significantly decreasing the solution cost. Finally, comparing policies in Table 5.2 shows that a more streamlined policy helps prevent stockouts. However, in both cases the average cost of the proactive policy is slightly higher than under the reactive policy.

Table 5.2: Summary of computational results for the Dynamic and Stochastic Inventory-Routing Problem on the seasonal data set

Transshipment	Instance	Reactive policy			Proactive policy		
	size	Solution	Time (s)	Avg. lost	Solution	Time (s)	Avg. lost
No	small ($n < 50$)	15994.92	0.1	0.41	14510.22	48.4	0.00
	medium ($50 \leq n \leq 100$)	40953.04	5.5	0.38	41071.74	499.7	0.00
	large ($n > 100$)	70442.24	758.3	0.41	73252.94	4734.7	0.00
Average		42463.40	254.6	0.40	42944.97	1760.9	0.00
Yes	small ($n < 50$)	15515.02	0.1	0.00	14160.04	48.1	0.00
	medium ($50 \leq n \leq 100$)	39164.04	5.8	0.00	40433.81	501.1	0.00
	large ($n > 100$)	66093.51	751.5	0.00	68918.08	4739.1	0.00
Average		40257.52	252.5	0.00	41170.64	1762.8	0.00

Finally, we provide the results for the base case defined with the correlated data set in Table 5.3. When demands are correlated and peaks occur simultaneously, emergency transshipments are still a powerful tool to mitigate lost demand, relocating inventory and making the system more robust, yet decreasing the average solution values. The use of forecasts helps reduce routing costs and stockouts.

Table 5.3: Summary of computational results for the Dynamic and Stochastic Inventory-Routing Problem on the correlated data set

Transshipment	Instance	Reactive policy			Proactive policy		
	size	Solution	Time (s)	Avg. lost	Solution	Time (s)	Avg. lost
No	small ($n < 50$)	15546.15	0.1	0.43	16466.03	48.3	0.00
	medium ($50 \leq n \leq 100$)	42940.79	14.4	0.48	40867.58	503.8	0.00
	large ($n > 100$)	75067.20	1506.5	0.47	71152.13	4727.4	0.00
Average		44518.05	507.0	0.46	42828.58	1759.8	0.00
Yes	small ($n < 50$)	15132.41	0.2	0.00	14822.28	48.2	0.00
	medium ($50 \leq n \leq 100$)	40526.86	15.4	0.00	40224.01	502.9	0.00
	large ($n > 100$)	70536.52	1749.1	0.00	68844.68	4713.8	0.00
Average		42065.26	588.2	0.00	41296.99	1755.0	0.00

Tables 5.1−5.3 show the solution values produced by the proactive policies are sometimes worse than those generated by the reactive policies, especially on large instances. A possible explanation is that the algorithm developed for the reactive policies solves the routing problem exactly, whereas the one proposed for the proactive policies relies on the ALNS matheuristic to sequence the customers. Even though this heuristic has been shown in earlier studies to provide good solutions (Coelho et al., 2012b,a), this time the number of customers is much larger. In particular, the *large* instances push the algorithm to its limit, and the ALNS implementation is streamlined to be executed several times in a rolling horizon fashion, which could explain the decrease in the solution quality. We further analyze the effect of running the ALNS algorithm in Section 5.5.3.2.

In addition to the analyses presented so far, we have investigated a number of other scenarios using the best of the proposed policies, i.e. the one described in Section 5.3.2.2.

5.5.3.2 Increasing the number of ALNS iterations

We first analyze the quality of the solutions obtained for the problem solved at each period when the ALNS heuristic is allowed to perform twice the original number of iterations, thus also roughly doubling the execution time. We now allow the ALNS to iterate 40,000 times. Average solutions for the proactive policy without and with transshipments are shown in Table 5.4. We see that allowing more computing time improves the average solution cost. For the case without transshipments, improvements are on average larger than 5%. This shows that these policies perform well if the algorithm used to solve the problem at each period is able to identify high quality solutions.

Table 5.4: Summary of solutions when applying the OU inventory policy for the Dynamic and Stochastic Inventory-Routing Problem on the standard data set with longer ALNS iterations

Instance	Without transshipments				With transshipments			
	Solution	Time (s)	Avg. lost	Increase (%)	Solution	Time (s)	Avg. lost	Increase (%)
small ($n < 50$)	9131.45	67.1	0.10	−7.20	8355.69	67.4	0.05	−2.16
medium ($50 \leq n \leq 100$)	30137.81	888.3	0.00	−4.39	26891.26	880.5	0.00	−4.23
large ($n > 100$)	60051.36	9248.9	0.00	−3.76	55530.30	9196.0	0.00	−2.08
Average	33106.87	3401.4	0.03	−5.12	30259.08	3381.3	0.02	−2.82

5.5.3.3 Applying an ML inventory policy

We have also implemented an ML inventory policy which relaxes the OU rule. Under this policy, the ALNS heuristic optimizes the quantities delivered while respecting the vehicle and the customer capacities. A summary of results on the standard data set is provided in Table 5.5. Specifically, we compute the average cost savings with respect to the OU policy when such a policy is applied, as well as the average lost demand (per customer per period) both without and with transshipments. Applying the ML policy yields reductions in solution costs and in lost demands.

Table 5.5: Summary of cost savings when applying the ML inventory policy for the Dynamic and Stochastic Inventory-Routing Problem on the standard data set

Instance	Without transshipments				With transshipments			
	Solution	Time (s)	Avg. lost	% increase over OU	Solution	Time (s)	Avg. lost	% increase over OU
small ($n < 50$)	10225.93	46.3	0.24	−0.78	7926.71	44.6	0.16	−9.96
medium ($50 \leq n \leq 100$)	30360.66	452.7	0.01	−1.50	26527.05	444.1	0.01	−3.78
large ($n > 100$)	61250.17	3860.1	0.01	−0.49	54292.38	4100.1	0.00	−4.19
Average	33945.59	1453.0	0.09	−0.92	29582.05	1529.6	0.06	−5.97

5.5.3.4 Varying the inventory holding costs

The inventory holding cost parameters play an important role in changing the balance between making more frequent deliveries or holding higher average inventories. To this end, we have analyzed two different scenarios: one in which inventory holding costs are doubled, and another in which they are halved. We present in Table 5.6 the results of these experiments. For all situations tested the variations occurred as expected, exhibiting a positive correlation between the inventory holding cost and the solution cost. Moreover, multiplying or dividing the inventory cost by two does not change the conclusion that the proactive policy still performs better than the reactive one.

Table 5.6: Summary of cost savings when varying the inventory holding costs for the Dynamic and Stochastic Inventory-Routing Problem on the standard data set

Inventory	Instance	Reactive policy				Proactive policy			
cost		Solution	Time (s)	Avg. lost	% increase	Solution	Time (s)	Avg. lost	% increase
Halved	small ($n < 50$)	14036.64	0.1	0.00	-2.52	8011.02	47.3	0.08	-8.29
	medium ($50 \leq n \leq 100$)	39703.27	18.9	0.00	-4.86	30366.73	721.7	0.00	-6.72
Average		26869.95	9.5	0.00	-3.69	19188.87	384.5	0.04	-7.51
Doubled	small ($n < 50$)	15074.69	0.1	0.00	4.66	9291.39	47.4	0.05	8.26
	medium ($50 \leq n \leq 100$)	45346.88	19.3	0.00	8.45	36087.193	707.3	0.00	10.62
Average		30210.79	9.7	0.00	6.55	22689.29	377.4	0.02	9.44

5.5.3.5 Increasing the length f of the planning horizon

We now evaluate the impact on the final solution cost of using a larger planning horizon. To this end, we have doubled to six the length f of the horizon used in the forecasts and in the ALNS, and we have solved a subset of instances from the standard data set. The fact that the ALNS matheuristic has to make twice as many decisions should be taken into account. In other words, keeping the number of ALNS iterations fixed, solution quality degradation is most likely to occur when doubling the length of the horizon. As a result, it makes sense to apply the idea used in Section 5.5.3.2 which consists of running the ALNS over a longer number of iterations. The average cost increases (or savings, when negative) are shown in Table 5.7. As expected, solution quality deteriorates with a longer horizon and computation times approximately double. Horizons of less than three periods make little sense since the main advantage of the proactive policy is to plan ahead and avoid visits to the same geographical area over consecutive periods, which is unlikely when f is very small.

Table 5.7: Summary of the impact on cost when time slices f are doubled (to six periods) under an OU inventory policy for the Dynamic and Stochastic Inventory-Routing Problem on the standard data set

Instance	Without transshipments				With transshipments			
	Solution	Time (s)	Avg. lost	% increase over OU	Solution	Time (s)	Avg. lost	% increase over OU
small ($n < 50$)	15553.77	62.8	0.04	0.25	10322.57	63.1	0.02	0.09
medium ($50 \leq n \leq 100$)	45963.58	1337.8	0.00	0.22	38864.04	1341.1	0.00	0.16
Average	30758.67	700.3	0.02	0.24	24593.31	702.1	0.01	0.12

5.5.3.6 Varying the service level

The percentage of the unknown demand covered against stockouts also plays an important role in the decision making process. We have varied the service level parameter, which directly affects the safety stock level of the proactive policy. We have run the algorithm on a subset of instances with a service level equal to 90% and to 99%, and we have summarized the results in Table 5.8. As the table shows, a lower service level means that customers are more likely to face a stockout, translating into increased transshipment costs; on the other hand, a higher service level protects customers against demand variations and emergency transshipments are then no longer needed as often, thus decreasing the total solution cost.

Table 5.8: Summary of cost savings when varying the service level for the Dynamic and Stochastic Inventory-Routing Problem on the standard data set

Instance	Low service level ($1-\alpha = 90\%$)				High service level ($1-\alpha = 99\%$)			
	Solution	Time (s)	Avg. lost	% increase	Solution	Time (s)	Avg. lost	% increase
small ($n < 50$)	8774.46	47.0	0.07	3.57	8145.81	47.4	0.03	−10.77
medium ($50 \leq n \leq 100$)	32257.77	702.2	0.00	−0.48	33566.23	738.6	0.00	2.55
Average	20516.12	374.6	0.03	1.54	20856.02	393.0	0.01	−4.11

5.5.3.7 Implementing consistency features in a dynamic environment

From a business and practical perspective, the decision making process is not only driven by costs but by quality of customer service. Our analysis has so far focused on cost minimization, disregarding other factors which may affect quality of service. Some of these factors were studied by Coelho et al. (2012b) who have analyzed the effect of incorporating different consistency features into IRP solutions. For example, it may be undesirable to dispatch an almost empty vehicle, or one would not like to frequently deliver small amounts to the same customer since this is time consuming for both parties. To this end, we have run a subset of instances subject to two consistency features next described.

We first apply the *vehicle filling rate* consistency feature ensuring that the vehicle is only used if it is at least γ% full, under the policy described in Section 5.3.2.2. We have tested the ML inventory policy with γ equal to 30, 50 and 70. Table 5.9 provides the average cost increase and the average lost demand (per customer per period) with respect to the base case. Running times are highly stable and, in general, as the requirement for the vehicle load increases, so does the solution cost. Adding this requirement to a deterministic environment (Coelho et al., 2012b) did not produce an increase of this magnitude.

Second, we apply a *quantity consistency* feature requiring that a customer can be visited only if the quantity delivered to it is at least twice its average demand. Results provided in Table 5.10 show that this policy yields a significant average cost

Table 5.9: Summary of the analysis for the Dynamic and Stochastic Inventory-Routing Problem with the vehicle filling rate consistency on the standard data set

Instance	$\gamma = 30$				$\gamma = 50$				$\gamma = 70$			
	Solution	Time (s)	Avg. lost	% increase	Solution	Time (s)	Avg. lost	% increase	Solution	Time (s)	Avg. lost	% increase
medium ($50 \leq n \leq 100$)	41233.80	707.8	0.00	31.06	41680.54	731.4	0.00	31.39	42474.15	744.6	0.00	34.22

increase with respect to the base case, and with respect to the average lost demand (per customer per period). Once more, ensuring consistent solutions over time turns out to be very costly in a dynamic environment, even though computational times are practically unchanged. Moreover, a slight increase in the average lost demand is observed when the quantities delivered to the customers are somewhat restricted.

Table 5.10: Summary of the analysis for the Dynamic and Stochastic Inventory-Routing Problem with the quantity consistency feature on the standard data set

Instance set	Under quantity consistency			
	Solution	Time (s)	Avg. lost	% increase in cost
medium ($50 \leq n \leq 100$)	48892.78	702.0	0.26	53.38

5.5.4 Final remarks

The two main features analyzed in these chapter are now summarized. First, the use of demand forecasts has proved a powerful asset for the solution of the DSIRP. However, it requires the use of an optimization algorithm that can sometimes take very long to run if high quality solutions are expected. Nevertheless, our implementation of the ALNS as a means of solving each periodic problem has proved to be very efficient and flexible in the sense that we have solved the problem under two inventory policies and with two consistency features.

The second option considered in this chapter concerns the use of lateral transshipments. Even if there are relatively few stockouts when transshipments are not considered, allowing them further reduces stockouts as well as the total cost. From an algorithmic point of view, enabling transshipments does not make the problem more difficult to solve since these can easily be integrated within the min-cost network flow problem which is used to compute the delivery quantities.

We have also analyzed the cost breakdown into its routing, inventory, direct deliveries and transshipments, and penalty components. Corroborating our preliminary findings from Sections 5.5.3.1 and 5.5.3.2, we found that routing costs are significantly reduced under proactive policies. This is due to the fact that when forecasts are used, the algorithm can avoid consecutive and costly visits to the same geographical area, yielding a better equilibrium between routing and inventory costs, in addition to reducing the use of emergency deliveries.

Finally, it is important to note that thanks to our choice of policies and to the algorithm design, the solution quality does not deteriorate when instances with very long horizons are solved. If a 20-period instance were to be solved by dynamic or stochastic programming, it is likely that it would be intractable, which is not the case for the rolling horizon algorithms we have developed.

5.6 Conclusions

We have successfully solved the dynamic and stochastic version of the IRP under different policies. The first one uses a reactive framework, in which future visiting decisions are based only on the current state of the inventory of the customers. We have also implemented a more involved policy under which demand forecasts are used to support future decisions. In both cases, we have solved the problem without and with lateral transshipments as a means of reducing lost demand and diminishing total costs. We have implemented these policies in a rolling horizon fashion. We have shown through extensive computational experiments that the algorithms proposed perform very well and allow the proactive policies to take advantage of stochastic information in the form of demand forecasts. We have shown that increasing the length of the rolling horizon does not have a positive impact on the overall solution quality. In contrast, increasing the computation time of the subproblem associated with each period significantly improves solution quality. We have analyzed the impact of different inventory holding costs and service levels. Our experiments have shown that solution costs are correlated with the inventory holding cost for all policies. Imposing a high service level ensures that customers are protected against demand variations, which avoids unnecessary emergency transshipments and reduces lost demand. Decreasing the service level even slightly negatively impacts the solution cost. Moreover, we have considered the inclusion of consistency features in the solutions of the DSIRP. Our experiments show that ensuring consistent solutions over time under a dynamic and stochastic environment is much more expensive than under a deterministic setting.

Chapter 6

Conclusions

In this thesis we have introduced, modeled and solved several types of inventory-routing problems. In particular, we have identified opportunities for increased flexibility and consistency within these problems, and have developed heuristic and exact algorithms for their solution. In the next paragraphs we outline our main findings as well as suggestions for future research.

We have proposed a comprehensive literature review in Chapter 2. The history of the inventory-routing problem, which dates back from 1983, was presented, along with a number of variants of the problem, their motivations, applications and solution procedures. However, the flexibility and consistency issues, which are the subject of this thesis, have not yet been addressed in any systematic way.

We have introduced the inventory-routing problem with transshipment in Chapter 3. Transshipments allow the supplier to streamline its distribution system, while granting the customers the advantage of sharing inventory, and thus reducing the risk of stockout. Our analysis was conducted on the single-vehicle case. We have developed an exact branch-and-cut algorithm which was applied to a mixed integer linear programming formulation of the problem. Since this algorithm ceases to be practical for large instances, we have also developed a powerful matheuristic for the same problem. This heuristic works in two steps. It first applies an adaptive large neighborhood search (ALNS) procedure to create vehicle routes. These are then used as inputs to an exact min-cost network flow problem which computes quantities to be delivered to each customer as well as transshipment origins and destinations.

In Chapter 4 we have incorporated six different consistency features into the IRP, and we have extended our analysis to the multi-vehicle version of the problem. These consistency features lead to solutions with higher quality of service for the customers, and more balanced vehicle loads for the supplier. They relate to the quantities delivered, the frequency of the deliveries and the workforce management.

We have again proposed an exact algorithm and a heuristic for these variants. The exact algorithm is an extension of the branch-and-cut method developed in Chapter 3. It can handle all proposed consistency features, either as constraints or as a penalty in the objective function. The heuristic is an extension of the ALNS procedure of Chapter 3 in which two subproblems are solved exactly. The first is the min-cost network flow problem for the computation of quantities delivered, and the second computes better upper bounds on the solution cost through the solution of a mixed integer linear program. All consistency features were implemented within the ALNS operators and in the network flow algorithm, the combination of which has proved to be highly flexible and efficient.

Finally, in Chapter 5 we have combined the two main ideas developed in this thesis and we have applied them to a context in which not all information is available when decisions are made. However, some probabilistic knowledge on future demands is available. This leads to a stochastic and dynamic inventory-routing problem. We have proposed four different policies for its solution, depending on whether demand forecasts are used in the solution process and on whether transshipments are allowed. All policies work within a rolling horizon framework in which smaller problems are solved iteratively whenever new information becomes available.

We view the main scientific contributions of this thesis as the introduction, modeling and optimization of meaningful variants of the classical IRP which allows for more flexible and consistent solutions. Flexibility enlarges the solution space, which yields lower cost solutions. Consistency leads to more realistic yet more costly solutions through the incorporation of restrictions into the problem. However, we have shown that the cost increase resulting from the introduction of consistency features tends to be relatively low. This study is also the first to investigate the benefits of incorporating forecasts and flexibility features within a dynamic and stochastic setting. From a methodological point of view, we have increased the scope and size of instances that can be solved optimally and we have developed heuristics which can often compute solutions with a measurable and high degree of accuracy. We believe our work opens up new research avenues and provides decision makers with better tools to handle some complex distribution management problems.

This thesis also contains some limitations. As in many operations research studies, we have made a number of assumptions to simplify the problem and make it tractable from a modeling and computational aspect. For instance, in most parts of this thesis we have ignored the stochasticity present in travel times, demand, and available vehicle capacity. From a computational perspective, large instances remain difficult to solve, even for powerful heuristic methods. This clearly calls for better algorithms.

We have identified a number of meaningful extensions to this thesis. The most interesting extension is probably the multi-product version of the inventory-routing problem. Modeling this problem would require a large increase in the number of variables, parameters and constraints, probably making it significantly harder to solve. We believe the ALNS framework put forward in this thesis can serve as a starting point as it was shown to be extremely flexible and it can handle such an extended model with relatively small modifications. Second, a number of other VRP extensions also make sense in the context of the IRP. One of these, in line with the concept of consistency, is the imposition of a constraint requiring customers to be visited at approximately the same time over different periods. This differs from the classical time windows setting in that one would not impose time windows for visits, but the optimization process would be free to determine visit times to the same customers, as long as they do not vary too much over time. Finally, multi-mode IRPs also represent a rich research area given that distribution is often carried out with company owned and leased vehicles, or with various combinations of transportation modes, some of which have fixed schedules and some are more flexible (see, e.g. Moccia et al. (2011)). The range of options available in this context is clearly very wide.

The IRP was introduced approximately 30 years ago and has since evolved into a rich research area. We believe this thesis has helped fill some gaps in this body of knowledge and will stimulate other researchers to pursue the study of this fascinating field.

Appendix A

Electronic appendix

In this appendix we provide electronic files with results for the full computational experiments carried out in this thesis. All files can be downloaded at `http://www.leandro-coelho.com/instances/thesis`.

A.1 Full results for the Inventory-Routing Problem with Transshipment

In this section we provide links to the files containing the data set and detailed solution values for all instances of the IRPT.

`http://www.leandro-coelho.com/instances/thesis/irpt/`

A.2 Full results for the Consistent Multi-Vehicle Inventory-Routing Problem

In this section we provide links to the files containing the data sets and detailed solution values for all instances of the Basic Multi-Vehicle IRP and of the Consistent MIRP.

`http://www.leandro-coelho.com/instances/thesis/consistent-mirp/`

A.3 Full results for the Exact Solutions of Inventory-Routing Problems

In this section we provide links to the files containing detailed exact solution values for all instances of the IRP.

`http://www.leandro-coelho.com/instances/thesis/exact_irp/`

A.4 Full results for the Dynamic and Stochastic Inventory-Routing Problem

In this section we provide links to the files containing detailed solution values for all instances of the DSIRP.

`http://www.leandro-coelho.com/instances/thesis/dsirp/`

References

T. F. Abdelmaguid. *Heuristic approaches for the integrated inventory distribution problem.* Ph.D. dissertation, University of Southern California, Los Angeles, 2004.

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

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

D. Adelman. Price-directed replenishment of subsets: Methodology and its application to inventory routing. *Manufacturing & Service Operations Management*, 5(4):348–371, 2003.

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

Y. Adulyasak, J.-F. Cordeau, and R. Jans. Formulations and branch-and-cut algorithms for multi-vehicle production and inventory routing problems. Technical Report G-2012-14, GERAD, Montreal, Canada, 2012.

E.-H. Aghezzaf. Robust distribution planning for the supplier-managed inventory agreements when demand rates and travel times are stationary. *Journal of the Operational Research Society*, 59(8):1055–1065, 2008.

E.-H. Aghezzaf, B. Raa, and H. van Landeghem. Modeling inventory routing problems in supply chains of high consumption products. *European Journal of Operational Research*, 169(3):1048–1063, 2006.

D. Aksen, O. Kaya, F. Salman, and Y. Akça. Selective and periodic inventory routing problem for waste vegetable oil collection. *Optimization Letters*, 6(6): 1063–1080, 2012.

M. Albareda-Sambola, E. Fernández, and G. Laporte. A computational comparison of several models for the exact solution of the capacity and distance constrained plant location problem. *Computers & Operations Research*, 38(8): 1109–1116, 2011.

S. Allen. Residtribution of total stock over several user locations. *Naval Research Logistics Quarterly*, 5(4):337–345, 1958.

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

A. Angulo, H. Nachtmann, and M. A. Waller. Supply chain information sharing in a vendor managed inventory partnership. *Journal of Business Logistics*, 25 (1):101–120, 2004.

S. Anily and A. Federgruen. One warehouse multiple retailer systems with vehicle routing costs. *Management Science*, 36(1):92–114, 1990.

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

C. Archetti, L. Bertazzi, G. Paletta, and M. G. Speranza. Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 12(38):1731–1746, 2011.

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

S. Axsäter. Modelling emergency lateral transshipments in inventory systems. *Management Science*, 36:1329–1338, 1990.

S. Axsäter. Forecasting. In *Inventory control*, volume 90 of *International Series in Operations Research & Management Science*, pages 7–42. Springer, New York, 2006.

F. Baita, W. Ukovich, R. Pesenti, and D. Favaretto. Dynamic routing-and-inventory problems: A review. *Transportation Research Part A: Policy and Practice*, 32(8):585–598, 1998.

J. F. Bard and N. Nananukul. Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering*, 57 (3):713–723, 2009.

J. F. Bard and N. Nananukul. A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, 37(12):2202–2217, 2010.

J. F. Bard, L. Huang, P. Jaillet, and M. Dror. Decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science*, 32 (2):189–203, 1998.

C. A. Barlett and S. Ghoshal. Building competitive advantage through people. *MIT Sloan Management Review*, 43(2):34–41, 2002.

M. Barrat. Positioning the role of collaborative planning in grocery supply chains. *International Journal of Logistics Management*, 14(2):53–66, 2003.

P. Bartodziej, U. Derigs, D. Malcherek, and U. Vogel. Models and algorithms for solving combined vehicle and crew scheduling problems with rest constraints: an application to road feeder service planning in air cargo transportation. *OR Spectrum*, 31(2):405–429, 2009.

D. O. Bausch, G. G. Brown, and D. Ronen. Scheduling short-term marine transport of bulk products. *Maritime Policy & Management*, 25(4):335–348, 1998.

B. M. Beamon. Measuring supply chain performance. *International Journal of Operations & Production Management*, 19(3):275–292, 1999.

J. E. Beasley. Fixed routes. *Journal of the Operational Research Society*, 35: 49–55, 1984.

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

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

G. Berbeglia, J.-F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1):8–15, 2010.

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

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

L. Bertazzi and M. G. Speranza. Continuous and discrete shipping strategies for the single link problem. *Transportation Science*, 36(3):314–325, 2002.

L. Bertazzi and M. G. Speranza. Matheuristics for inventory routing problems. In J. R. Montoya-Torres, A. A. Juan, L. H. Huatuco, J. Faulin, and G. L. Rodriguez-Verjan, editors, *Hybrid Algorithms for Service, Computing and Manufacturing Systems: Routing and Scheduling Solutions*, pages 1–14. IGI Global, Hershey, PA, 2012.

L. Bertazzi, M. G. Speranza, and W. Ukovich. Minimization of logistic costs with given frequencies. *Transportation Research Part B: Methodological*, 31(4): 327–340, 1997.

L. Bertazzi, G. Paletta, and M. G. Speranza. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36(1):119–132, 2002.

L. Bertazzi, G. Paletta, and M. G. Speranza. Minimizing the total cost in an integrated vendor-managed inventory system. *Journal of Heuristics*, 11(5-6): 393–419, 2005.

L. Bertazzi, M. Savelsbergh, and M. G. Speranza. Inventory routing. In B. Golden, S. Raghavan, E. Wasil, R. Sharda, and S. Voß, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces Series*, pages 49–72. Springer, 2008.

L. Bertazzi, A. Bosco, F. Guerriero, and D. Laganà. A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies*, 2012. doi: 10.1016/j.trc.2011.06.003.

R. Bhatnagar and C.-C. Teo. Role of logistics in enhancing competitive advantage: A value chain framework for global supply chains. *International Journal of Physical Distribution & Logistics Management*, 39(3):202–226, 2009.

D. E. Blumenfeld, L. D. Burns, J. D. Diltz, and C. F. Daganzo. Analyzing trade-offs between transportation, inventory and production costs on freight networks. *Transportation Research Part B: Methodological*, 19(5):361–380, 1985.

M. Boudia and C. Prins. A memetic algorithm with dynamic population management for an integrated production-distribution problem. *European Journal of Operational Research*, 195(3):703–715, 2009.

L. D. Burns, R. W. Hall, D. E. Blumenfeld, and C. F. Daganzo. Distribution strategies that minimize transportation and inventory costs. *Operations Research*, 33(3):469–490, 1985.

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

A. M. Campbell, L. Clarke, A. J. Kleywegt, and M. W. P. Savelsbergh. The inventory routing problem. In T. G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 95–113. Springer, Boston, 1998.

M. W. Carter, J. M. Farvolden, G. Laporte, and J. Xu. Solving an integrated logistics problem arising in grocery distribution. *INFOR*, 34(4):290–306, 1996.

P. Chandra and M. L. Fisher. Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503–517, 1994.

X. Chen, G. Hao, X. Li, and K. F. C. Yiu. The impact of demand variability and transshipment on vendor's distribution policies under vendor managed inventory strategy. *International Journal of Production Economics*, 139(1): 42–48, 2012.

T. W. Chien, A. Balakrishnan, and R. T. Wong. An integrated inventory allocation and vehicle routing problem. *Transportation Science*, 23(2):67–76, 1989.

M. Christiansen. Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science*, 33(1):3–16, 1999.

M. Christiansen and K. Fagerholt. Robust ship scheduling with multiple time windows. *Naval Research Logistics*, 49(6):611–625, 2002.

M. Christiansen and B. Nygreen. A method for solving ship routing problems with inventory constraints. *Annals of Operations Research*, 81:357–378, 1998a.

M. Christiansen and B. Nygreen. Modelling path flows for a combined ship routing and inventory management problem. *Annals of Operations Research*, 82:391–412, 1998b.

M. Christiansen and B. Nygreen. Robust inventory ship routing by column generation. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 197–224. Springer, New York, 2005.

M. Christiansen, K. Fagerholt, and D. Ronen. Ship routing and scheduling: Status and perspectives. *Transportation Science*, 38(1):1–18, 2004.

M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Maritime transportation. In C. Barnhart and G. Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 189–284. North-Holland, Amsterdan, 2007.

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

N. Christofides and J. E. Beasley. The periodic routing problem. *Networks*, 14 (2):237–256, 1984.

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

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

L. C. Coelho, J.-F. Cordeau, and G. Laporte. The inventory-routing problem with transshipment. Technical Report CIRRELT-2011-21, Montreal, Canada, 2011a.

L. C. Coelho, J.-F. Cordeau, and G. Laporte. Consistency in multi-vehicle inventory-routing. Technical Report CIRRELT-2011-66, Montreal, Canada, 2011b.

L. C. Coelho, J.-F. Cordeau, and G. Laporte. The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537–2548, 2012a.

L. C. Coelho, J.-F. Cordeau, and G. Laporte. Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, 24(1):270–287, 2012b.

M. A. Cohen, P. R. Kleindorfer, and H. L. Lee. Optimal stocking policies for low usage items in multi-echelon inventory systems. *Naval Research Logistics Quarterly*, 33:17–38, 1986.

J.-F. Cordeau, G. Laporte, M. W. P. Savelsbergh, and D. Vigo. Vehicle routing. In C. Barnhart and G. Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 367–428. North-Holland, Amsterdan, 2007.

M. Dada. A two-echelon inventory system with priority shipments. *Management Science*, 38:1140–1153, 1992.

A. F. Darrat and M. Zhong. On testing the random-walk hypothesis: A model-comparison approach. *Financial Review*, 35(3):105–124, 2000.

C. Das. Supply and redistribution rules for twp-location inventory systems: one period analysis. *Management Science*, 21:765–776, 1975.

S. Dauzère-Pérès, A. Nordli, A. Olstad, K. Haugen, U. Koester, M. P. Olav, G. Teistklub, and A. Reistad. Omya Hustadmarmor optimizes its supply chain for delivering calcium carbonate slurry to European paper manufacturers. *Interfaces*, 37(1):39–51, 2007.

M. Desrochers and G. Laporte. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36, 1991.

B. Dezső, A. Jüttner, and P. Kovács. LEMON - an open source C++ graph template library. *Electronic Notes in Theoretical Computer Science*, 264(5): 23–45, 2011.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

E. B. Diks and A. G. de Kok. Controlling a divergent two-echelon network with transshipments using the consistent appropriate share rationing policy. *International Journal of Production Economics*, 45:369–379, 1996.

R. Dondo, C. A. Méndez, and J. Cerdá. The supply-chain pick-up and delivery problem with transshipments. In J. Jeżowski and J. Thullie, editors, *19th European Symposium on Computer Aided Process Engineering*, volume 26 of *Computer Aided Chemical Engineering*, pages 1009–1014. Springer, Heidelberg, 2009.

M. Dror and M. O. Ball. Inventory/routing: Reduction from an annual to a short-period problem. *Naval Research Logistics*, 34(6):891–905, 1987.

M. Dror and L. Levy. A vehicle routing improvement algorithm comparison of a "greedy" and a matching implementation for inventory routing. *Computers & Operations Research*, 13(1):33–45, 1986.

M. Dror, M. O. Ball, and B. L. Golden. A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research*, 4 (1-4):3–23, 1985.

D. Eddelbuettel and R. François. *RInside: C++ classes to embed R in C++ applications*, 2012. URL `http://CRAN.R-project.org/package=RInside`. R package version 0.2.6.

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

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

F. Erhun and P. Keskinocak. Collaborative supply chain management. In K. G. Kempf, P. Keskinocak, and R. Uzsoy, editors, *Planning Production and Inventories in the Extended Enterprise*, volume 151 of *International Series in Operations Research & Management Science*, pages 233–268. Springer, New York, 2011.

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

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

M. Fischetti, J. J. Salazar-González, and P. Toth. Experiments with a multicommodity formulation for the symmetric capacitated vehicle routing problem. In *Proceedings of the 3rd Meeting of the EURO Working Group on Transportation*, pages 169–173, Barcelona, Spain, 1995.

M. L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle-routing. *Networks*, 11(2):109–124, 1981.

P. Francis, K. Smilowitz, and M. Tzur. The periodic vehicle routing problem and its extensions. In B. L. Golden, S. Raghavan, and E. A. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 239–261. Springer, New York, 2008.

F. Fumero and C. Vercellis. Synchronized development of production, inventory and distribution schedules. *Transportation Science*, 33(3):330–340, 1999.

G. Gallego and D. Simchi-Levi. On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer r-systems. *Management Science*, 36(2): 240–243, 1990.

G. Gallego and D. Simchi-Levi. Rejoinder to 'A note on bounds for direct shipping costs'. *Management Science*, 40(10):1393, 1994.

M. J. Geiger and M. Sevaux. Practical inventory routing: A problem definition and an optimization method. In C. Artner, K. F. Doerner, R. F. Hartl, and F. Tricoire, editors, *Proceedings of the EU/MEeting: Workshop on Client-Centered Logistics and International Aid*, pages 32–35, Vienna, 2011a.

M. J. Geiger and M. Sevaux. On the use of reference points for the biobjective Inventory Routing Problem. In S. Ceschia, L. Di Gaspero, M. Loghi, A. Schaerf, and T. Urli, editors, *Proceedings of the MIC 2011: The IX Metaheuritics International Conference*, pages 141–149, Udine, 2011b.

G. Ghiani, F. Guerriero, G. Laporte, and R. Musmanno. Real-time vehicle routing: Solutions concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 151(1):1–11, 2003.

F. Glover. Multi-start and strategic oscillation methods – principles to exploit adaptive memory. In M Laguna and J. L. González-Velarde, editors, *OR Computing Tools for Modeling, Optimization and Simulation - Interfaces in Computer Science and Operations Research*, pages 1–24. Kluwer, Boston, 2000.

M. Goetschalckx. *Supply Chain Engineering*, volume 161 of *International Series in Operations Research & Management Science*. Springer, New York, 2011.

A. V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms*, 22(1):1–29, 1997.

B. L. Golden, A. A. Assad, and R. Dahl. Analysis of a large-scale vehicle-routing problem with an inventory component. *Large Scale Systems in Information and Decision Technologies*, 7(2–3):181–190, 1984.

Y. Gong and E. Yücesan. Stochastic optimization for transshipment problems with positive replenishment lead times. *International Journal of Production Economics*, 135(1):61–72, 2012.

P. Goodwin, D. Önkal, and M. Thomson. Do forecasts expressed as prediction intervals improve production planning decisions? *European Journal of Operational Research*, 205(1):195–201, 2010.

C. Groër, B. L. Golden, and E. A. Wasil. The consistent vehicle routing problem. *Manufacturing & Service Operations Management*, 11(4):630–643, 2009.

R. Grønhaug, M. Christiansen, G. Desaulniers, and J. Desrosiers. A branch-and-price method for a liquefied natural gas inventory routing problem. *Transportation Science*, 44(3):400–415, 2010.

D. Gross. Centralized inventory control in multilocation supply systems. In H. E. Scarf, D. M. Gilford, and M. W. Shelly, editors, *Multistage inventory models and techniques*, chapter 3, pages 47–84. Stanford University Press, Stanford, 1963.

R. W. Hall. A note on bounds for direct shipping cost. *Management Science*, 38(8):1212–1214, 1992.

Y. T. Herer and A. Rashit. Lateral stock transshipments in a two-location inventory system with fixed and joint replenishment costs. *Naval Research Logistics*, 46(5):525–547, 1999.

Y. T. Herer and R. Roundy. Heuristic for one-warehouse multiretailer distribution problem with performance bounds. *Operations Research*, 45(1):102–115, 1997.

Y. T. Herer, M. Tzur, and E. Yücesan. Transshipments: An emerging inventory recourse to achieve supply chain leagility. *International Journal of Production Economics*, 80(3):201–212, 2002.

Y. T. Herer, M. Tzur, and E. Yücesan. The multilocation transshipment problem. *IEE Transactions*, 38(3):185–200, 2006.

M. Hewitt, G. L. Nemhauser, and M. W. P. Savelsbergh. Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. *INFORMS Journal on Computing*, 22(2):314–325, 2010.

S.-H. Huang and P.-C. Lin. 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, 2010.

L. M. Hvattum and A. Løkketangen. Using scenario trees and progressive hedging for stochastic inventory routing problems. *Journal of Heuristics*, 15 (6):527–557, 2009.

L. M. Hvattum, A. Løkketangen, and F. Glover. New heuristics and adaptive memory procedures for boolean optimization problems. In J. Karlof, editor, *Integer Programming - Theory and Practice*, pages 1–18. CRC Press, Boca Raton, FL, 2005.

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

R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3):1–22, 2008.

R. J. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder. *Forecasting with Exponential Smoothing: the State Space Approach*. Springer-Verlag, Berlin, 2008.

R. J. Hyndman, S. Razbash, and D. Schmidt. *forecast: Forecasting functions for time series and linear models*, 2012. URL `http://CRAN.R-project.org/package=forecast`. R package version 3.19.

A. A. Javid and N. Azad. Incorporating location, routing and inventory decisions in supply chain network design. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):582–597, 2010.

H. Jönsson and E. A. Silver. Analysis of a two-echelon inventory control system with complete redistribution. *Management Science*, 33:215–227, 1987.

I. Kara, G. Laporte, and T. Bektas. A note on the lifted Miller-Tucker-Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158(3):793–795, 2004.

U. Karmarkar and N. Patel. The one-period, n-location distribution problem. *Naval Research Logistics Quarterly*, 24(4):559–575, 1977.

A. J. Kleywegt, V. S. Nori, and M. W. P. Savelsbergh. The stochastic inventory routing problem with direct deliveries. *Transportation Science*, 36(1):94–118, 2002.

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

K. Krishnan and V. Rao. Inventory control in n warehouses. *Journal of Industrial Engineering*, 16(3):212–215, 1965.

G. Laporte. Fifty Years of Vehicle Routing. *Transportation Science*, 43(4): 408–416, 2009.

G. Laporte, R. Musmanno, and F. Vocaturo. An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands. *Transportation Science*, 44(1):125–135, 2010.

H. L. Lee. A multi-echelon inventory model for repairable items with emergency lateral transshipments. *Management Science*, 33:1302–1316, 1987.

H. L. Lee and W. Seungjin. The whose, where and how of inventory control design. *Supply Chain Management Review*, 12(8):22–29, 2008.

F. Li, B. L. Golden, and E. A. Wasil. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34(9):2734–2742, 2007.

J. Li, H. Chen, and F. Chu. Performance evaluation of distribution strategies for the inventory routing problem. *European Journal of Operational Research*, 202(2):412–419, 2010.

S.-C. Liu and W.-T. Lee. A heuristic method for the inventory routing problem with time windows. *Expert Systems with Applications*, 38(10):13223–13231, 2011.

S. G. Makridakis, S. C. Wheelwright, and R. J. Hyndman. *Forecasting: Methods and Applications.* Wiley, New York, 1998.

V. Maniezzo, T. Stützle, and S. Voß. *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming.* Springer, New York, 2009.

M. Mateo, E.-H. Aghezzaf, and P. Vinyes. A combined inventory routing and game theory approach to solve a real-life distribution problem. *International Journal of Business Performance and Supply Chain Modelling*, 4(1): 75–89, 2012.

A. Mercer and X. Tao. Alternative inventory and distribution policies of a food manufacturer. *Journal of the Operational Research Society*, 47(6):755–765, 1996.

S. Michel and F. Vanderbeck. A column-generation based tactical planning method for inventory routing. *Operations Research*, 60(2):382–397, 2012.

A. S. Minkoff. A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. *Operations Research*, 41(1):77–90, 1993.

B. K. Mishra and S. Raghunathan. Retailer- vs. vendor-managed inventory and brand competition. *Management Science*, 50(4):445–457, 2004.

L. Moccia, J.-F. Cordeau, G. Laporte, S. Ropke, and M. P. Valentini. Modeling and solving a multimodal transportation problem with flexible-time and scheduled services. *Networks*, 57(1):53–68, 2011.

N. H. Moin and S. Salhi. Inventory routing problems: a logistical overview. *Journal of the Operational Research Society*, 58(9):1185–1194, 2007.

N. H. Moin, S. Salhi, and N. A. B. Aziz. An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. *International Journal of Production Economics*, 133(1):334–343, 2011.

L. M. Nonås and K. Jörnsten. Heuristics in the multi-location inventory system with transshipments. In H. Kotzab, S. Seuring, M. Müller, and G. Reiner, editors, *Research Methodologies in Supply Chains Management*, pages 509–524. Physica-Verlag, Heidelberg, 2005.

L. M. Nonås and K. Jörnsten. Optimal solution in the multi-location inventory system with transshipments. *Journal of Mathematical Modelling and Algorithms*, 6(1):47–75, 2007.

D. L. Olson and M. Xie. A comparison of coordinated supply chain inventory management systems. *International Journal of Services and Operations Management*, 6(1):73–88, 2010.

Ö. Özer. Inventory management: Information, coordination, and rationality. In K. G. Kempf, P. Keskinocak, and R. Uzsoy, editors, *Planning Production and Inventories in the Extended Enterprise*, volume 151 of *International Series in Operations Research & Management Science*, pages 321–365. Springer, New York, 2011.

M. W. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1): 60–100, 1991.

C. Paterson, G. Kiesmüller, R. Teunter, and K. Glazebrook. Inventory models with lateral transshipments: A review. *European Journal of Operational Research*, 210(2):125–136, 2011.

A.-S. Pepin, G. Desaulniers, A. Hertz, and D. Huisman. A comparison of five heuristics for the multiple depot vehicle scheduling problem. *Journal of Scheduling*, 12(1):17–30, 2009.

J. A. Persson and M. Göthe-Lundgren. Shipment planning at oil refineries using column generation and valid inequalities. *European Journal of Operational Research*, 163(3):631–652, 2005.

V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia. A review of dynamic vehicle routing problems. Technical Report CIRRELT-2011-62, Montreal, Canada, 2011.

D. Pisinger and S. Ropke. A general heuristic for the vehicle routing problem. *Computers & Operations Research*, 34(8):2403–2435, 2007.

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

C. Prins, C. Prodhon, P. Soriano, A. Ruiz, and R. Wolfler Calvo. Solving the capacitated location routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic. *Transportation Science*, 41(4):470–483, 2007.

H. N. Psaraftis. Dynamic vehicle routing problems. In B. L. Golden and A. A. Assad, editors, *Vehicle Routing: Methods and Studies*, pages 223–248. North-Holland, Amsterdam, 1998.

W. W. Qu, J. H. Bookbinder, and P. Iyogun. An integrated inventory-transportation system with modified periodic policy for multiple products. *European Journal of Operational Research*, 115(2):254–269, 1999.

Y. Qu and J. F. Bard. A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research*, 39(10):2439–2456, 2012.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL `http://www.R-project.org`.

B. Raa and E.-H. Aghezzaf. Designing distribution patterns for long-term inventory routing with constant demand rates. *International Journal of Production Economics*, 112(1):255–263, 2008.

B. Raa and E.-H. Aghezzaf. A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research*, 192(2):429–441, 2009.

N. Ramkumar, P. Subramanian, T. Narendran, and K. Ganesh. Mixed integer linear programming model for multi-commodity multi-depot inventory routing problem. *OPSEARCH*, Forthcoming, 2012. doi: 10.1007/s12597-012-0087-0.

R. Ribeiro and H. R. Lourenço. Inventory-routing model for a multi-period problem with stochastic and deterministic demand. Technical Report 275, Department of Economics and Business, Universitat Pompeu Fabra, 2003.

L. W. Robinson. Optimal and approximate policies in multiperiod, multi-location inventory models with transshipments. *Operations Research*, 38(2):278–295, 1990.

D. Ronen. Ship scheduling: The last decade. *European Journal of Operational Research*, 71(3):325–333, 1993.

D. Ronen. Marine inventory routing: shipments planning. *Journal of the Operational Research Society*, 53(1):108–114, 2002.

S. Ropke and D. Pisinger. An adaptive large neighborghood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006a.

S. Ropke and D. Pisinger. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3): 750–755, 2006b.

R. Roundy. 98%-effective integer-ratio lot-sizing for one-warehouse multi-retailer systems. *Management Science*, 31(11):1416–1430, 1985.

N. Rudi, S. Kapur, and D. Pyke. A two-location inventory model with transshipment and local decision making. *Management Science*, 47(12):1668–1680, 2001.

B. Satır, S. Savasaneril, and Y. Serin. Pooling through lateral transshipments in service parts systems. *European Journal of Operational Research*, 220(2): 370–377, 2012.

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

P. Shaw. A new local search algorithm providing high quality solutions to vehicle routing problems. Technical report, University of Strathclyde, Glasgow, 1997.

Q. Shen, F. Chu, and H. Chen. A Lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. *Computers & Chemical Engineering*, 35(10):2113–2123, 2011.

D. Simchi-Levi, X. Chen, and J. Bramel. *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics and Supply Chain Management.* Springer-Verlag, New York, 2005.

S. Sindhuchao, H. E. Romeijn, E. Akçali, and R. Boondiskulchok. An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity. *Journal of Global Optimization*, 32(1):93–118, 2005.

K. Smilowitz, M. Nowak, and T. Jiang. Workforce management in periodic delivery operations. *Transportation Science*, Forthcoming, 2012. doi: 10.1287/trsc.1120.0407.

O. Solyalı and H. Süral. A single supplier-single retailer system with an order-up-to level inventory policy. *Operations Research Letters*, 36(5):543–546, 2008.

O. Solyalı and H. Süral. 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, 2011.

O. Solyalı, J.-F. Cordeau, and G. Laporte. Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327–340, 2012.

J. H. Song and K. C. Furman. A maritime inventory routing problem: Practical approach. *Computers & Operations Research*, 2012. doi: 10.1016/j.cor.2010.10.031.

SPEC. Standard Performance Evaluation Corporation. `http://www.spec.org/`. Accessed August, 2011.

M. G. Speranza and W. Ukovich. Minimizing transportation and inventory costs for several products on a single link. *Operations Research*, 42(5):879–894, 1994.

M. G. Speranza and W. Ukovich. An algorithm for optimal shipments with given frequencies. *Naval Research Logistics*, 43(5):655–671, 1996.

J. Stacey, M. Natarajarathinam, and C. Sox. The storage constrained, inbound inventory routing problem. *International Journal of Physical Distribution & Logistics Management*, 37(6):484–500, 2007.

M. Stålhane, J. G. Rakke, C. R. Moe, H. Andersson, M. Christiansen, and K. Fagerholt. A construction and improvement heuristic for a liquefied natural gas inventory routing problem. *Computers & Industrial Engineering*, 62(1):245–255, 2012.

N. Suakkaphong and M. Dror. Managing decentralized inventory and transshipment. *Top*, 19(2):480–506, 2011.

G. Tagaras. Effects of pooling on the optimization and service levels of two-location inventory systems. *IIE Transactions*, 21:250–257, 1989.

G. Tagaras. Pooling in multi-location periodic inventory distribution systems. *Omega*, 27(1):39–59, 1999.

G. Tagaras and M. A. Cohen. Pooling in two-location inventory systems with non-negligible replenishment lead times. *Management Science*, 38:1067–1083, 1992.

C. D. Tarantilis, E. E. Zachariadis, and C. T. Kiranoudis. A hybrid meta-heuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):255–271, 2009.

S. Tayur. Computing optimal stock levels for common components in an assembly system. *GSIA Working Papers*, 1995.

P. Trudeau and M. Dror. Stochastic inventory-routing − route design with stockouts and route failures. *Transportation Science*, 26(3):171–184, 1992.

K. T. Uggen, M. Fodstad, and V. S. Nørstebø. Using and extending fix-and-relax to solve maritime inventory routing problems. *TOP*, Forthcoming, 2011. doi: 10.1007/s11750-011-0174-z.

S. Viswanathan and K. Mathur. Integrating routing and inventory decisions in one warehouse multiretailer multiproduct distribution system. *Management Science*, 43(3):294–312, 1997.

K. E. Wee and M. Dada. Optimal policies for transshipping inventory in a retail network. *Management Science*, 51(10):1519–1533, 2005.

M. Wen, J.-F. Cordeau, G. Laporte, and J. Larsen. The dynamic multi-period vehicle routing problem. *Computers & Operations Research*, 37(9):1615–1623, 2010.

R. Wolfler Calvo and N. Touati-Moungla. A matheuristic for the dial-a-ride problem. In J. Pahl, T. Reiners, and S. Voß, editors, *Network Optimization*, volume 6701 of *Lecture Notes in Computer Science*, pages 450–463. Springer, Berlin/Heidelberg, 2011.

Y. G. Yu, F. Chu, and H. X. Chen. A note on coordination of production and distribution planning. *European Journal of Operational Research*, 177(1): 626–629, 2007.

Y. G. Yu, H. X. Chen, and F. Chu. A new model and hybrid approach for large scale inventory routing problems. *European Journal of Operational Research*, 189(3):1022–1040, 2008.

Q.-H. Zhao, S.-Y. Wang, and K. K. Lai. A partition approach to the inventory/routing problem. *European Journal of Operational Research*, 177(2): 786–802, 2007.

Q.-H. Zhao, S. Chen, and C.-X. Zang. Model and algorithm for inventory/routing decisions in a three-echelon logistics system. *European Journal of Operational Research*, 19(3):623–635, 2008.

H. Zhong, R. W. Hall, and M. M. Dessouky. Territory planning and vehicle dispatching with driver learning. *Transportation Science*, 41(1):74–89, 2007.

L. Zou, M. Dresner, and R. Windle. A two-location inventory model with transshipments in a competitive environment. *International Journal of Production Economics*, 125(2):235–250, 2010.