



HAL
open science

Modelling and solving complex vehicle routing problems with integrated management of shared inventories in supply chains

Fatima Ezzahra Achamrah

► **To cite this version:**

Fatima Ezzahra Achamrah. Modelling and solving complex vehicle routing problems with integrated management of shared inventories in supply chains. Business administration. Université Paris-Saclay, 2022. English. NNT: 2022UPAST063 . tel-04010781

HAL Id: tel-04010781

<https://theses.hal.science/tel-04010781v1>

Submitted on 2 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modelling and solving complex vehicle
routing problems with integrated
management of shared inventories in
supply chains

*Modélisation et résolution de problèmes complexes de
tournées de véhicules et gestion intégrée des stocks
partagés dans les chaînes logistiques*

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 573 Interfaces : matériaux, systèmes, usages
(Interfaces)

Spécialité de doctorat: Génie Industriel

Graduate School : GS Sciences de l'ingénierie et des systèmes, Référent
: CentraleSupélec

Thèse préparée dans l'unité de recherche Laboratoire Génie Industriel
(Université Paris-Saclay, CentraleSupélec), sous la direction de Fouad RIANE,
professeur, la co-direction de Evrin SAHIN, professeure

Thèse présentée et soutenue à Paris-Saclay, le 07 Juin 2022, par

Fatima Ezzahra ACHAMRAH

Composition du jury

Marija Jankovic

Professeure des Universités, CentraleSupélec

Hamid Allaoui

Professeur, Université d'Artois

Mohamed Zied Babai

Professeur, KEDGE Business School

Sabine Limbourg

Professeure, HEC-Université de Liège

Fouad Riane

Professeur, Ecole Centrale Casablanca

Evren Sahin

Professeure, CentraleSupélec

Présidente du jury

Rapporteur & Examineur

Rapporteur & Examineur

Examinatrice

Directeur de thèse

Directrice de thèse

Titre: Modélisation et résolution de problèmes complexes de tournées de véhicules et gestion intégrée des stocks partagés dans les chaînes logistiques.

Mots clés: gestion des chaînes logistiques, stocks partagés, optimisation, modélisation, métaheuristiques, machine learning

Résumé: La thèse porte sur le développement de modèles de décision pour la gestion des stocks partagés, intégrée à l'optimisation des tournées de véhicules dans les chaînes logistiques. Nous nous concentrons plus particulièrement sur le partage des stocks de produits finis et des supports réutilisables de transport (SRTs) qui circulent en boucle fermée entre les partenaires d'une chaîne logistique. Pour mener cette recherche, nous partons d'études de cas pratiques, décrivant la réalité du partage des stocks. Nous nous intéressons particulièrement à la distribution d'articles de mode et de pièces de rechange pour les produits finis et à la gestion des SRTs dans les industries automobile et agroalimentaire tout en exploitant le paradigme de l'Internet Physique. Ce dernier est un concept novateur d'encapsulation des marchan-

disés dans des objets intelligents qui s'inspire de la capacité d'Internet à interconnecter des réseaux hétérogènes pour le transposer aux réseaux logistiques. Nous étudions la littérature connexe pour analyser les travaux existants et mettre clairement en avant nos contributions. Nous développons pour chaque type de stock partagé des modèles mathématiques spécifiques, et des méthodes de résolution originales basées sur l'hybridation de la modélisation mathématique, des algorithmes co-évolutifs et de l'apprentissage par renforcement profond. Des expérimentations sont menées pour évaluer la pertinence des modèles proposés, la performance des approches de résolution, et pour analyser les résultats obtenus afin de produire des recommandations pertinentes.

Title: Modelling and solving complex vehicle routing problems with integrated management of shared inventories in supply chains.

Keywords: supply chains management, inventory sharing, optimisation, modelling, metaheuristics, machine learning

Abstract: The thesis focuses on the development of decision models for managing inventory sharing, integrated with vehicle routing optimisation in supply chains. Specifically, we focus on inventory sharing of finished products and returnable transport items (RTIs) that flow in a closed loop between partners in supply chains. To conduct this research, we start from practical case studies, describing the reality of inventory sharing. We are particularly interested in the distribution of fashion items and spare parts for finished goods and in the management of RTIs in the automotive and food industries while exploiting the paradigm of

the Physical Internet. We review the related literature to analyse existing work and clearly highlight our contributions. We develop for each type of shared inventory specific mathematical models, and original solution methods based on the hybridisation of mathematical modeling, co-evolutionary algorithms and deep reinforcement learning. Experiments are conducted to evaluate the relevance of the proposed models, the performance of the resolution approaches, and to analyse the results obtained in order to produce relevant recommendations.



Modelling and solving complex vehicle routing problems
with integrated management of shared inventories in supply
chains

Final deposit

Thesis National Number (NNT): 2022UPAST063

By: **Fatima Ezzahra Achamrah** ^{a,b}

Supervised by: **Fouad Riane** ^{a,b}, **Evren Sahin** ^b

^aComplex Systems and Interactions, Ecole Centrale of Casablanca

^bLaboratoire Genie Industriel, CentraleSupélec, Paris Saclay University

Acknowledgement

I would like to express my sincere gratitude to my academic supervisors. First, I would like to thank professor Fouad Riane for his guidance, care, and insightful discussions. I am also grateful to professor Evren Sahin for sharing her knowledge and expertise in conducting this research. Then, I express my sincere gratitude to professor Sabine Limbourg for her insights into the industrial context and stakes, as well as her advice, patience, and trust.

I would like to thank the jury members, professors Hamid Allaoui, Marija Jankovic, and Mohamed Zied Babai for agreeing to review my thesis manuscript and allowing me to discuss my research with them.

I extend my sincere thanks to professor Christine Di Martinelly and professor El-Houssaine Aghez-zaf for following this research with great interest and care.

This journey was made particularly rich and enjoyable thanks to the academic and administrative staff of Ecole Centrale Casablanca and CentraleSupélec. Mainly, I would like to extend my sincere thanks, in alphabetic order, to Ali Attajer, Anas Bouchnita, Bernard Yannou, Carole Stoll, Delphine Martin, Jakob Puchinger, Ouidad Benhlime, Suzanne Thurán, and others.

Above all, I owe a huge debt of gratitude to my mother. Nothing, including this PhD, would have been possible if it hadn't been for her. This PhD and all of my previous and future "successes" are dedicated to her.

Abstract

This thesis is a part of a joint project in industrial engineering and complex systems, known as the "Pôle Commun de Recherche CentraleSupélec/EC Casablanca", between the Complex Systems and Interactions research group of Ecole Centrale Casablanca and Laboratoire Génie Industriel of CentraleSupélec. It concerns modelling and solving complex problems for optimising vehicle routing and integrated management of shared inventory in supply chains. Sharing inventory concerns finished products and empty reusable transport items (RTIs) that flow in a closed loop (e.g., pallets and containers). Sharing inventory is authorised between entities belonging to the same supply chain echelon to reduce logistics costs and enhance service levels at customers' locations. We started with practical case studies to conduct this research, highlighting the need to promote inventory sharing. We were particularly interested in the distribution of fashion items, perishable products and spare parts for finished goods and in managing RTIs in the automotive and food industries. Second, we reviewed the related literature to analyse existing work and position our contributions. Then, we developed specific mathematical models for each type of shared inventory and original solving methods based on the hybridisation of mathematical modelling, co-evolutionary algorithms and deep reinforcement learning. Finally, we conducted experiments to evaluate the proposed models' relevance and the resolution approaches' performance and to analyse the results obtained to produce relevant recommendations.

Keywords: supply chains management, inventory sharing, optimisation, modelling, metaheuristics, machine learning

Résumé Général

Cette thèse s'inscrit dans le cadre du partenariat en génie industriel et systèmes complexes, appelé "Pôle Commun de Recherche CentraleSupélec/EC Casablanca", entre l'unité de recherche Systèmes Complexes et Interactions de l'Ecole Centrale Casablanca et le Laboratoire Génie Industriel de CentraleSupélec. Elle porte sur la modélisation et la résolution de problèmes complexes pour l'optimisation des tournées des véhicules et la gestion intégrée des stocks partagés dans les chaînes logistiques. Les stocks partagés concernent à la fois les produits finis et les supports réutilisables de transport (SRTs) vides qui circulent en boucle fermée (e.g., les palettes et les conteneurs). Le partage des stocks est autorisé entre les entités appartenant au même échelon de la chaîne logistique afin de réduire les coûts logistiques et d'améliorer le niveau de service chez les clients.

Pour mener cette recherche, nous nous sommes basés sur l'étude de cas pratiques, mettant en évidence la nécessité de promouvoir le partage des stocks. Nous nous sommes particulièrement intéressés à la distribution d'articles de mode, de produits périssables et de pièces de rechange pour les produits finis, ainsi qu'à la gestion des SRTs dans les industries automobile et alimentaire. Ensuite, nous avons passé en revue la littérature connexe afin d'analyser les travaux existants et de positionner clairement nos contributions.

Nous avons proposé des approches de gestion pour les acteurs de la chaîne logistique désireux de collaborer dans le but de réduire les coûts tout en maintenant un niveau de service élevé. Plus spécifiquement, nous avons élaboré des approches de gestion des stocks et des tournées de véhicules consistant à promouvoir le multi-sourcing pour atténuer les pénuries. Cette politique combine les livraisons régulières du(es) producteur(s) aux clients (e.g., dépôts, points de vente, usines), et le partage des stocks. L'autorisation de la substitution des produits est également investiguée.

D'un point de vue développement mathématique, nous avons proposé des modèles mettant en exergue le partage des stocks, la collecte et la livraison couplées à l'optimisation des tournées de véhicules, ainsi que les substitutions de produits. Ces formulations prennent également en compte différents profils de demandes et prise de décision centralisée et décentralisée, ainsi que la logistique inverse.

En ce qui concerne les approches de résolution, nous avons conçu des algorithmes intégrant les connaissances acquises des modèles, et de l'hybridation de métaheuristiques sophistiquées. Étant donné que la recherche aléatoire conduit à une évolution lente et à une faible efficacité de convergence des métaheuristiques, nous avons utilisé l'apprentissage par renforcement profond pour construire des algorithmes avec une forte auto-adaptabilité et des performances axées sur les connaissances et les objectifs. Les métaheuristiques proposées ont été comparées aux algorithmes exacts et métaheuristiques bien connus en termes de qualité de solution et du temps de calcul.

Enfin, nous avons évalué et quantifié l'impact de la mise en avant du multi-sourcing sur la performance globale de la chaîne logistique. Nous avons démontré, au moyen des expérimentations, que le multi-sourcing présente un apport vital pour une amélioration de la résilience et de l'efficacité dans des contextes de plus en plus volatiles, complexes, incertains et ambigus.

Mots clés: gestion des chaînes logistiques, stocks partagés, optimisation, modélisation, métaheuristiques, machine learning

Contents

1	General Introduction	11
1.1	Supply chain trends and related issues	11
1.2	Research scope definition	13
1.3	Research motivation	14
1.4	Theoretical Background	17
1.4.1	IRP Problems	17
1.4.2	Transshipment	18
1.4.3	Substitution	20
1.4.4	Pickups and deliveries and transshipment Price	20
1.4.5	Returnable Transport Items in sharing context	21
1.5	Research questions	22
1.6	Research methodology and dissertation organisation	24
1.6.1	Research methodology	24
1.6.2	Dissertation organisation	26
2	Modelling and solving a deterministic multi-product multi-vehicle IRP-T	31
2.1	Introduction	32
2.1.1	IRP and transshipment	33
2.1.2	Pickup and delivery	35
2.1.3	Scientific contribution and organisation of the paper	36
2.2	Problem setting	37
2.3	Mathematical formulation	38
2.4	Resolution Approach	43
2.4.1	Constructive phase	44
2.4.2	Improvement phase	46
2.5	Computational experiments and discussion	50
2.5.1	Experiment design and instances	50
2.5.2	Parameters setting of the matheuristic	52
2.5.3	Computational experiments	53
2.6	Conclusions & Perspectives	59
3	Modeling and solving a stochastic multi-product multi-vehicle IRP-T with product substitution	61
3.1	Introduction	62
3.2	Related work	66
3.3	Problem setting	67
3.4	Mathematical formulation	68
3.4.1	Mathematical modelling	68

3.4.2	Sample Average Approximation	71
3.5	Computational experiments	74
3.5.1	Experimental design	74
3.5.2	Computational experiments	76
3.6	Conclusions & Perspectives	80
4	Modeling and solving a dynamic and stochastic multi-product multi-vehicle IRP- T with substitutions	83
4.1	Introduction	84
4.2	Related work	86
4.2.1	Inventory Routing Problem	86
4.2.2	IRP with uncertainty	87
4.2.3	IRP with transshipment	88
4.2.4	Paper main contributions	89
4.3	Problem setting	90
4.4	Solution policy	91
4.4.1	Routing model	92
4.4.2	Transshipment and substitution model	94
4.5	Genetic Algorithm and Deep Reinforcement Learning	96
4.5.1	Genetic Algorithm	96
4.5.2	Deep Reinforcement	97
4.6	Computational Results	99
4.6.1	Experimental design	99
4.6.2	Parameters tuning	100
4.6.3	Computational results for the single-vehicle-product DSIRP with and without transshipment	100
4.6.4	Computational results for the multi-vehicle-product DSIRP with substitution and transshipment	101
4.6.5	Computational results for other demand patterns	102
4.7	Conclusions & Perspectives	104
5	Modeling and solving a decentralised and deterministic multi-product multi- vehicle VRPT	105
5.1	Introduction	106
5.2	Related work	108
5.2.1	Pickup and Delivery Problem	108
5.2.2	Bi-level optimisation	109
5.3	Mathematical formulation	110
5.3.1	Bi-level integer program	110
5.3.2	Reformulation of LL model using Karush-Kuhn-Tucker conditions	112
5.4	Resolution approach	113
5.4.1	Hybrid Genetic Algorithm coupled with reinforcement learning	113
5.5	Computational experiments	116

5.5.1	Data set and parameters tuning	116
5.5.2	Primary insights into ϵ and β optimal value	117
5.5.3	Results on small and large instances	117
5.6	Conclusions & perspectives	123
6	Modeling and solving a deterministic multi-vehicle IRP-T with pickup and delivery of shared RTIs	125
6.1	Introduction	126
6.2	Related Work	129
6.3	Mathematical Formulation	132
6.3.1	Mathematical Model for IRPPDS	132
6.3.2	Mathematical Model for DM	137
6.3.3	Mathematical Model for SM	138
6.4	Resolution Approach	141
6.4.1	Artificial Immune System	142
6.4.2	AIS enhanced with Deep Q-Learning	144
6.5	Implementation and Experimental Analysis	146
6.6	Computational Experiments	148
6.6.1	Results on Small Instances Solved Using CPLEX	148
6.6.2	First Insights into the Effectiveness of the Resolution Approach on Small Instances	151
6.6.3	Extra Experiments on Large Instances Solved Using GA-DQL and AIS-DQL	152
6.6.4	Sensitivity Analysis on Unit Cost	153
6.7	Conclusions and Perspectives	155
7	Modeling and solving a stochastic multi-supplier multi-hub multi-customer IRP-T with shared inventory sharing of finished products and RTIs	157
7.1	Introduction	158
7.2	Related work	161
7.3	Problem description	162
7.4	Problem formulation	164
7.5	Resolution approach	167
7.5.1	Artificial immune system	167
7.5.2	AIS enhanced by Deep Q-learning	169
7.6	Computational experiments	170
7.6.1	Experimental design	170
7.6.2	Parameters tuning of AIS and DQ	171
7.6.3	Computational results	172
7.7	Conclusions & perspectives	180
8	General conclusions	183
	Appendices	209

A Computational results for the classical IRP (Chapter 2)	211
B Computational results for different models and demand distributions under consideration (Chapter 3)	219

1 - General Introduction

In this introductory section, we present the supply chain trends and related issues. We then define the research scope, and provide the research motivations and the theoretical background. Later, we present the related research questions and methodology, and report our main contributions. Finally, we give a synoptic view of the thesis structure.

1.1 . Supply chain trends and related issues

Supply chains are continually changing, and the logistics environment is becoming exceedingly challenging. Globalisation, along with technologies, is driving this transformation (Faruquee et al., 2021). Indeed, products reach customers from companies involved in various processes and activities to produce and provide value through upstream and downstream linkages (Dudukalov et al., 2021). Moreover, customers behavioural patterns are rapidly shifting as they are increasingly connected, endowed with an increased capacity to act through networked exchanges, making digital their daily use (Dash and Chakraborty, 2021). Nowadays, it has never been so easy to consume. Whether in a few clicks, via phone or on a street corner, customers will always find a way to see what they need (Reinartz et al., 2019). Customers are thus more likely to change their source of supply and to opt for a similar product from a different supplier who promises to deliver faster the right quantities of product to the right place at the most reasonable cost.

With these significant disruptions, companies must be on top of their game. They are being forced to transform and re-invent their operations management to meet their customers' needs, preserve, and develop their market positioning and growth (Reinartz et al., 2019). This is particularly true today with the current COVID crisis severely straining companies (Agostino et al., 2021; Bai et al., 2021) and overturning all the theories that have been established around logistics, more precisely regarding inventory management and product availability (Shi et al., 2021). For instance, 94% of Fortune 1000 companies (e.g., Walmart, Boeing, Procter & Gamble, Coca-Cola, Amazon) are seeing supply chain disruptions from COVID, particularly product unavailability (Butt, 2021). In France, retailers lost €2.5 billion in potential sales, and in the UK, £20 billion in lost sales in non-food stores due to COVID (Panzone et al., 2021). In Japan, stocks of finished goods have dropped to levels not even seen in the wake of the earthquake and tsunami disaster of 2011 (Batth, 2021). Not to mention the \$1.1 trillion of total revenue lost

worldwide each year due to inventory distortion (IHL, 2015).

Packaging is also concerned with unavailability issues that impacted global trade during COVID. Indeed, almost all finished products – including clothes, spare parts, medicines, and processed food products – are shipped, stored, and handled using Returnable Transport Items (RTIs) like drums, containers, pallets, boxes, barrels, trolleys, refillable liquid, or gas containers (Kochańska et al., 2021). Along with the ever-growing globalised supply chains, RTIs have become more popular over the last decades as they eliminate wastes that one-way packaging may generate while conforming with the government's regulations for sustainable supply chains (Iassinovskaia et al., 2017; Limbourg et al., 2016; Glock, 2017). Furthermore, due to COVID, the RTIs market in the United States was valued at \$7,827.5 million in 2020 and is expected to reach \$ 11,040.06 million by 2026, at a CAGR of 5.9% over the forecast period (2021 - 2026) (Mordor Intelligence, 2021). On the other hand, to avoid packaging unavailability, many companies frequently tend to invest more into RTIs, resulting in higher holding costs (Meherishi et al., 2021)). Moreover, supply chain players experience loss rates of RTIs varying from 3 to 20% (TrackX, 2017). This mismanagement lengthens turnaround times and pushes players to overinvest in these assets leading to inefficient budgetary practices: companies buy new RTIs to replace the lost ones and recruit additional staff to handle them (Meherishi et al., 2021; Ullah et al., 2021).

These disruptions and changing conditions force companies to adopt different approaches to shipping products through their supply chains. The continuous flow of data regarding customers, purchases, deliveries, locations, and inventory makes product movement from various origins to various destinations intrinsically much more complex and dynamic than it used to be (Dudukalov et al., 2021). Since building customer loyalty is achieved through product availability, timeliness, and consistency of delivery, companies should properly integrate and understand the relevance of logistics activities. Moreover, they need to be more resilient and reactive to changes, to provide more effective responses to customers whose demand is becoming more difficult to predict and variable over time. This highlights inventory management and distribution optimisation problems by resurfacing a longstanding research area yet highly relevant to today's world: the inventory routing problem (IRP). Accordingly, this research project investigates the emerging issues of modelling and solving complex routing problems. It emphasises that appropriate solutions to such complex issues can be identified when supply chain players are jointly accountable for total system value creation, which can be accomplished when information and resources are shared, and when broader parts of the supply chain are together modelled and optimised.

1.2 . Research scope definition

We consider in our research project distribution networks where a set of products are shipped from one or different suppliers to different customers (e.g., retailers, Point of Sales, plants) to satisfy their demands that vary over time. The supplier-s and the customers incur inventory costs. Shipments are performed by an owned or outsourced vehicle fleet with a given capacity. Transportation costs occur for each movement of vehicles.

Products shipment from supplier-s to the customer can occur under two main configurations. In the first one, deliveries are triggered by customers placing orders. The delivery company (supplier-s) should then find for each day an efficient and effective combination of routes that minimises the distance travelled from its warehouse to the customers' local stores. This problem is called "Vehicle Routing Problem (VRP)", well-known in the literature in various forms with varying constraints (Laporte, 2009). In the second configuration, the delivery company manages the inventory replenishment of its customers. Under a supply chain agreement called "Vendor Managed Inventory (VMI)", the supplier-s and the customer manage inventory jointly and mutually, reducing the risk of shortage or oversupply of products (Archetti et al., 2007). The supplier-s can schedule the logistics operations more productively, yielding reductions in inventory costs and shortages. In the literature, this problem is referred to as Inventory Routing Problems (Coelho et al., 2012b; Coelho and Laporte, 2013a).

Yet, it is well known that both VRP and IRP are NP-Hard (Coelho and Laporte, 2013a; Laporte, 2009). They are notoriously challenging for exact methods (Desaulniers et al., 2016), but they can efficiently be tackled by combining metaheuristic and mathematical modelling techniques, referred to as matheuristics (Chitsaz et al., 2019). Indeed, the use of metaheuristics in most cases does not capture different aspects of real-world problems, leading to impractical resolution approaches (Archetti and Speranza, 2014). This keeps research in this area interesting and gives rise to practical problems for investigation, particularly when it comes to urban logistics.

The product's transportation and delivery and the management of their reusable transport supports are vital processes in different supply chains. It plays a decisive role in customer value creation through the provided service level and determines the business logistics costs. The current competitive environment and the development of new technologies have forced companies to share resources more than just information and practices. The scope of our research is then defined around partnerships, cooperation, and collaboration processes that suppliers and their customers need to develop to work together through new forms of distribution of goods to reduce costs while guaranteeing the service level to customers and preventing product shortages as much as possible. It also focuses on developing original and efficient resolution

approaches to handle the combinatorial complexity of the problems.

1.3 . Research motivation

Product shortage is a timely issue that reduces the supply chain's players' profits, harms product branding, and impedes the overall supply chain's sustainability. This is especially true regarding the uncertainties that characterise the international business landscape in which supply chains operate today. Supply chain managers are asked to secure supplies and avoid possible shortages. They must play a delicate balancing act: reducing stock levels to reduce costs while ensuring no shortages so that no interruption of production and delivery to customers may incur. This is not easy, especially if orders for a particular product category are unpredictably increasing. The question is how to proceed? One way is to authorise resource sharing, particularly lateral transshipment (LT), which relies on inventory sharing between members belonging to the same echelon of supply chains (Paterson et al., 2011; Coelho et al., 2012b; Lefever et al., 2018; Abouee-Mehrizi et al., 2015). This is a practice of inventory pooling used thanks to its potential to increase profitability and service levels through risk pooling. It is common to see LT between different stores nowadays as it allows faster deliveries compared to emergency orders and savings from unnecessary production, especially for perishable goods.

Many illustrations of this practice can be found in various industry sectors, such as automotive, spare parts, fashion, retailing, and different online trading platforms (Grahovac and Chakravarty, 2001; Abouee-Mehrizi et al., 2015; Accorsi et al., 2019; Zhao and Bisi, 2010). The advantages of inventory sharing are noticeable in terms of improved service levels and profitability. These advantages are ultimately determined by inventory holding and player sharing decisions. This requires academic research that examines how sharing decisions are made and quantifies savings through analytical models and simulation studies.

Avoiding shortages and improving service levels can result from a substitution strategy. In competitive markets, customers choose from various products according to their needs. They may choose to buy their preferred products or opt for substitute products in case of unavailability. Substitutes can lead to healthy market competition between products, which is in the customers' best interest, preventing a market monopoly. Indeed, substitution is another strategy to cope with shortages that can be combined with LT. Examples of such practices stem from perishable products like blood products (e.g., artificial blood that can be used as a substitute to mitigate the risks of blood transfusions) and spare parts (e.g., original equipment manufacturer parts that can be substituted by aftermarket parts called replacement or pat-

tern parts). In both cases, substitution can represent an alternative to meet customers' demands better, mainly if decision-makers are not fully aware of future events.

Our research is mainly motivated by real-life industrial applications. We investigated two Moroccan companies seeking a better way to supply their distribution networks via a Central Warehouse (CW). The objective is to demonstrate that LT can help reduce costs and enhance service levels. We started with a fashion distribution company that manufactures a set of finished products and allocates them via a CW to its customers (i.e., Point of Sales POS) according to a pre-established policy based on the previous sales of similar products. In most cases, as customer demands vary, the stock imbalance can regularly happen: some customers have over-stock (and therefore incur high holding cost), while others are out of stock (and therefore incur shortage cost: loss of sales). The company aims to increase its revenues and service level and avoid any loss of profit due to the non-availability of products at the right places and moments.

We also examined the case of an automotive spare parts distributor, which has been active for decades in the distribution of spare parts and automotive accessories sector. The distributor supplies its customers via a CW with original equipment manufacturer (OEM) and replacement or aftermarket parts, also called replacement parts or pattern parts (PP). The latter are reverse-engineered OEM parts, designed to perform the same way as OEM parts, are less expensive, have a quality equivalent to or better than OEM parts, and are provided with a wider variety. On the other hand, spare parts are known to be at the most significant risk of obsolescence and may collectively account for up to 60% of the total stock value (Johnston et al., 2003). Moreover, their demand pattern is intermittent, exhibiting an infrequent rate and extreme dispersal over periods which often hinders the reduction of lost sales at the customer's location.

Reducing the shortage of finished products at customers' locations by authorising multi-sourcing (i.e., LT and substitutions) is essential. On the other hand, the unavailability of packaging, particularly RTIs, hinders the performance of the supply chains. Indeed, such assets usually flow in a closed-loop supply chain between different players. Once received and unloaded at a given level of the supply chain, the empty RTI can either be collected and returned to the sender, or they can be reused by the receiver to ship his products and thus continue to flow downstream the supply chain. Therefore, there exist two flows of RTIs that must be managed: forward flows, which correspond to the forward distribution of finished products loaded on RTIs, and reverse flows, which correspond to the collection and return of empty RTIs to their owners. However, in many practical cases, the quantity of empty RTIs in inventory at the owner's location may not be sufficient to meet

the future demands of his customers (e.g., plants, POS, retailers). This may be due, for example, to delays in returning empty RTIs from customers, loss of RTIs, damage to the returned RTIs, or highly variable customer demands, making it challenging to meet customers' needs. This thesis focuses on a new management approach to overcome the shortcomings of managing empty RTI inventories. Specifically, we consider the case of a two-level closed-loop supply chain comprising a set of suppliers delivering the product to common customers. This can be, for instance, the case of stillages in an automotive supply chain, plastic crates in a Fresh-food supply chain, or pallets in the Fast Moving Consumer Goods (FMCG) supply chain. We assume that RTIs are "mutualised" throughout the supply chain. Consequently, suppliers can sidestep the shortage of empty RTIs at their levels and reduce the cost of transportation, inventory holding, and new RTIs procurement. Thus, we addressed a deterministic multi-supplier, multi-customer IRP with pickup and delivery of multi and shared RTIs.

To bring into the play the benefits of promoting inventory sharing of finished products and RTIs, we built up all developed models to investigate a three-level-closed loop supply chain in a Physical Internet (PI) setting. The latter is an innovative concept of encapsulating goods in intelligent objects, globally standard, smart, green, shared, and modular containers, inspired by the Internet's ability to interconnect heterogeneous networks and transpose them to logistics networks. In this thesis, we develop an optimisation model for inventory routing of reusable containers flowing in a closed loop. We investigate a PI-supply chain in which a set of suppliers deliver their products using RTIs to customers (e.g., plants, retailers, etc.). Direct and reverse flows of the RTIs are consolidated at the level of a set of PI-hubs. Such management is highly relevant in the FMCG industry, where products are non-durable, delivered in packaged form, at low prices and in high volumes, and frequently purchased. This thesis focuses on the inventory routing model under stochastic demand of reusable containers exploiting the PI concept, which is new to the literature. Second, it considers inventory sharing between the PI-hubs. Sharing includes both empty and loaded RTIs (i.e., finished products).

The analysis of the different configurations under consideration reveals a specific need in inventory management which may be common to several other companies. Indeed, companies seek alternative methods to manage inventory sharing by promoting product substitution and transshipment within the distribution network. The objective is to ensure a high customer service level commitment avoiding as much as possible loss of sales while minimising logistics costs. Our research work aims to answer these critical concerns. The research objective is to develop a modelling framework endowed with solving capabilities that allows capturing different sharing strategies and in-

investigating their potential benefits for the players.

1.4 . Theoretical Background

In this section, we present the theoretical background of the thesis. More in-depth literature reviews are to be found in the remaining chapters.

1.4.1 . IRP Problems

In its classical version, an IRP refers to a combination of inventory management, vehicle routing and delivery scheduling decisions. Such a problem arises in several industries and services under a VMI setting where a supplier/vendor is responsible for managing its customer's inventory. The supplier must deliver products to several geographically dispersed customers subject to side constraints (Coelho et al. (2014b)). He can reduce the overall costs of his activities to achieve a competitive advantage by integrating routing, inventory, and distribution decisions instead of optimising them independently.

The IRP has received considerable attention from researchers over the years, and multiple variants considering multiple attributes of the problem have been studied. IRP problems can be classified according to different criteria. The first is the number of customers and suppliers (Coelho et al., 2012a). We distinguish the one-to-many version where one supplier serves several customers (Bell et al., 1983; Burns et al., 1985; Abdelmaguid, 2004); and the many-to-many version, which is less investigated, with multiple customers being served by numerous suppliers (Christiansen, 1999; Ronen, 2002). Routing is the second criterion that distinguishes between the direct routing, where there is only one customer per route, and the multiple routing, where different customers are visited using the same route (Zhao et al., 2008). The inventory strategy pre-established to satisfy customers is the third classification criterion. The most used policies are the Maximum Level strategy (ML) and the Order-Up to level (OU) one. The replenishment level is flexible under an ML inventory strategy but is restricted by the resources available to each customer (Savelsbergh and Song, 2008; Coelho and Laporte, 2013b). Under an OU policy, the quantity delivered must fill its inventory capacity whenever a customer is visited. The characteristics of the vehicle fleet are another classification criterion. The number of vehicles can be set at one, set at many or unconstrained. The vehicles can be homogeneous, heterogeneous of limited capacity or unconstrained (Zhao et al., 2008; Coelho et al., 2012a). In most papers in the IRP literature, only one product is considered, whereas many VMI applications are concerned with multiple product distributions. Few papers address the multi-product inventory routing problem (Coelho et al., 2012a).

IRP can also be classified regarding the nature of demands. Static and deterministic IRP is the most studied. Unlike static and stochastic IRP in which demands distributions are known before planning, in a dynamic and stochastic IRP (DSIRP), demands are gradually revealed over time (Bertazzi et al., 2013). This is particularly true in the aerospace, IT, blood, and automotive industry contexts (Turrini and Meissner, 2019). In this case, the objective is not to deliver a static result, but a solution policy using the information revealed, outlining which measures need to be performed as time passes (Coelho et al., 2014a).

1.4.2 . Transshipment

Inventory and transportation are the most critical issues of the logistics system and the two main drivers that provide value to customers ensuring product availability, timeliness and consistency of delivery. Supply chain managers are asked to secure supplies and avoid possible shortages of product availability. When dealing with distribution networks with different retailers supplied from one or multiple warehouses, managers must constantly play a tricky balancing act, juggling between the need to reduce inventory levels to reduce costs and the need to avoid shortages not to impact the service levels promised to customers. This is not easy, significantly, if customer demand for a particular product category is unpredictably increasing. The question that arises then is how to proceed in such conditions. A possible solution could stem from resource sharing, particularly LT, which relies on inventory sharing between members belonging to the same echelon of a supply chain (Paterson et al., 2011; Coelho et al., 2012b; Lefever et al., 2018; Abouee-Mehrzi et al., 2015).

Unlike the "traditional" and "hierarchical" design of an inventory system, with transportation flows from one echelon to the next, i.e. from manufacturers/suppliers to customers (e.g., wholesalers, retailers), a flexible approach allows LT within an echelon, i.e. between customers (Paterson et al., 2011). In this way, members of the same echelon pool their inventories. LT can occur at predetermined times before all demand is realised or at any time to respond to shortages or potential shortages of products (Timajchi et al., 2019). LT concerns, for instance, perishable products (e.g., blood products), low demand items (e.g., spare parts), fast-moving items (e.g., packaged food) and empty RTIs inventory (Grahovac and Chakravarty, 2001; Abouee-Mehrzi et al., 2015; Accorsi et al., 2019).

LT is certainly not a new practice, but it has been brought to the fore during the pandemic thanks to its numerous advantages (Ekren et al., 2021):

- it allows faster response to customers compared to emergency deliveries in case these latter is significantly longer and more expensive (e.g., supplier's CW located far from customers) (Paterson et al., 2011).

- it re-balances the entire system's stock levels to react to scenarios where one of the locations faces a shortage while others have residual stock in hand. Thus, reducing the inventory distortion (shortages and overstocks) (Cavagnini et al., 2018).
- it reduces transportation and inventory costs at customers (Timajchi et al., 2019).
- it reduces excessive production, especially for perishable products, as the products that were to be manufactured and then shipped to a customer to respond to emergency orders will instead be replaced by the products held at other customers' locations (Dehghani et al., 2021).
- it saves vehicle resources, eases traffic congestion, and reduces environmental pollution (Mirzaei and Seifi, 2015).

LT can be encountered either in centralised or decentralised settings. In a centralised setting, a single decision-maker (e.g., supplier) aims to reduce the overall logistics costs at his level and customers' locations (Paterson et al., 2011). Minimising the total cost may include inventory holding at suppliers and customers, shortage, routing and transshipment costs. To address the planning of routing and LT, an inventory routing problem (IRP) with LT (IRPT) is solved (Coelho et al., 2012b; Azadeh et al., 2017). In this specific problem (IRPT), the decision-maker determines simultaneously when to deliver which products to which customer; how much to deliver to each customer; how much inventory to share and transship, and the routing of vehicles for regular shipment (e.g., from supplier's central warehouse to customers) and for LT (e.g., between customers).

In decentralised systems, each decision-maker (supplier and customers) operates to minimise his costs (Axsäter, 2003). Each LT policy has to balance conflicting interests and manage inventories and deliveries in this setting. Moreover, it has to choose the right LT price so that both supplier and customers would benefit from LT and its related cost would not exceed the profit to be made (Shao et al., 2011; Hezarkhani and Kubiak, 2010; Atan et al., 2018). The supplier's objective is to choose inventory levels and routes to be constructed either for regular shipment or LT. In addition, customers minimise their inventory holding and shortage costs; while satisfying constraints relative to vehicle routing (VRP) and product availability. Therefore, costs are often optimised locally as both players aim at reducing their objective functions that are narrowly defined. Accordingly, solving such problems requires hierarchical decision-making, which belongs to the multilevel optimisation family (Grahovac and Chakravarty, 2001).

1.4.3 . Substitution

Another management policy that relies on promoting multi-sourcing options to mitigate shortages without resorting to drastic inventory increases is using product substitutes (Hssini et al., 2016). In practice, more precisely in a multi-product configuration environment, when a supplier cannot guarantee the replenishment of product P , retailers, for instance, may suggest to their clients the use of substitute product S having the same functionality as P to meet his demand. This practice offers opportunities to increase service quality.

Suppliers can promote product substitution along with LT whenever demand exceeds the available stock at their customers' level by assuming that emergency supplies from the CW are significantly extended and expensive. That is, LT and substitution can be considered to meet expected demands with the use of the same part from the inventory of another retailer in the network (LT) or with the help of product substitutes held in their stock (substitution). This can be the case with perishable products such as blood products, where artificial blood can be used as a substitute to mitigate the risks of blood transfusions and shortage of supply (Hssini et al., 2016).

Along with LT, product substitution is most encountered in the spare parts supply chain, where aftermarket parts can substitute original spare parts. The typical studied configuration is a two-level supply chain configuration in which a company supplies its CW with OEM products and replacement, or aftermarket parts, also called replacement parts or pattern parts (PP) to a set of its customers. In such a case, shipments are assumed to be direct, if necessary, from the CW to any customer. This problem can generally be modelled as a multi-product IRPT with substitutions under static or dynamic stochastic demands. The objective is to minimise the total cost, including the costs of holding inventory, transportation, LT, substitution and lost sales. To the best of our knowledge, none of the existing papers incorporates product substitution within the settings and promotes LT between customers to avoid shortages.

1.4.4 . Pickups and deliveries and transshipment Price

When a supplier agrees with his customers to share inventories, he must decide, at each visit to a customer, the quantity to deliver and pick up. Pickups and deliveries of shared stocks can be either outsourced or integrated. In the first case, pickups and deliveries are performed by a third-party carrier that only takes charge of shipments of transshipped products between customers, while regular shipments from the supplier are performed using his vehicles. In this case, the decision-maker determines only the nodes and periods where LT may occur are determined, and inventories are managed so that LT can be performed (Coelho et al., 2012b). When pickups and de-

liveries can be carried out by the same vehicles that make regular shipments, LT should be integrated into the design of vehicles' routes.

Modelling inventory sharing brings out an essential concept of collaboration in decentralised supply chains. Indeed, each LT policy must balance conflicting interests to achieve economies of scale. Also, it must determine the right LT price such that both the company and its customers (i.e., POS, depots, and retailers) benefit from LT and the resulting costs do not exceed the profit to be made (Atan et al., 2018). This is the case of decentralised supply chains in which each decision-maker or level of the supply chain works to optimise its costs (Liao et al., 2020; Li et al., 2020). This problem is often modelled using game theory and considering the supplier acting as a Stackelberg leader (Axsäter, 2003). That is, customers (followers in such a game) optimise their objective function subject to the value of the leader variable. In such a model, each player's preliminary decision is to enhance the service level while maintaining a minimum total cost, including transportation, inventory, lost sales, and LT. That is, the objective of the supplier (upper level) is to choose inventory levels and routes to be constructed (according to his objective), knowing that the customers (lower level) will follow optimally; while satisfying a set of constraints relative to vehicle routing, products availability, and inventory management.

Moreover, players under a decentralised setting seek approaches that suggest trade-off solutions to manage their conflict of interests (Li et al., 2020). Indeed, as a part of the collaboration, the supplier and the customers may agree to incur each their own holding cost as well as a part of the cost of lost sales associated with the products shortage and a part of the cost of LT, more precisely when LT is not outsourced. The supplier incurs, in turn, the vehicle routing cost for regular shipments. Therefore, inventory sharing can only be attractive to all players if the LT and lost sales costs share are optimally defined. Such a configuration is thoroughly investigated in this thesis.

1.4.5 . Returnable Transport Items in sharing context

Product packaging is of great importance in the supply chain. It serves multiple functions that impact distribution, warehousing, business operations and customers' actions (Limbourg et al., 2016). We can distinguish between single-use and multi-use packaging or RTIs (Iassinovskaia et al., 2017). The various items used for transporting goods naturally differ per industry. However, almost all finished products are shipped, stored, and handled using Returnable Transport Items (RTIs) like drums, containers, pallets, boxes, barrels, trolleys, refillable liquid, or gas containers. These under-recognised packaging items ensure goods flow safely, cheaply, and efficiently through the entire supply chain. They are of significant value. Their prices can fluctuate many times a year under the pressure of new trends in e-commerce and

logistics, particularly since the COVID crisis ([Mordor Intelligence, 2021](#)).

RTIs unavailability hinders the flow synchronicity through the entire supply chain and impacts its global performance. Thus, reducing the shortage of RTIs at suppliers' locations by promoting sharing and authorising multi-sourcing is very important. Indeed, RTIs flow in a closed-loop supply chain between different players. Therefore, their forward (loaded RTIs) and reverse (empty RTIs) flows must be managed. However, suppliers often experience delays in returning empty RTIs from customers, loss of RTIs, damage to the returned RTIs, or highly variable customer demands, making it challenging to meet customers' needs ([Bortolini et al., 2018](#)). Therefore, managing such assets becomes a primary concern of supply chain managers in the same way as managing inventories, warehouses, machines, and vehicles. To avoid reusable packaging unavailability, many companies frequently tend to over-invest in RTIs, which results in higher inventory holding and purchasing costs. It is then of such interest to consider sharing RTIs among owners as a new management approach to overcome the shortcomings of managing empty RTIs inventories. This approach may concern, for instance, the case of stillages in an automotive supply chain, plastic crates in a Fresh-food supply chain, or pallets in the FMCG supply chain. This thesis contributes to modelling sharing RTIs within multi-level closed-loop supply chains.

As mentioned above, this research project is also driven by gaps identified in the literature. Literature reviews, conducted and presented in the following chapters, highlight that few papers consider the study of multi-product, multi-supplier VRP/IRPT with deterministic or stochastic demands, and in which lost sales due to shortages are viewed as a measure of customer satisfaction. Moreover, it can also be noted that most papers consider centralised decision making, a single actor responsible for managing the overall operations. Moreover, to the best of our knowledge, none of the existing papers incorporates product substitution within the settings and promoting LT between customers to avoid shortages. Furthermore, one can observe the lack of contributions regarding integrating the LT in the design of the vehicle routes, which are traditionally decomposed or separated as researchers assume that LT is always outsourced. Beyond the fact that multi-sourcing creates flexibility and mitigates the risks of supply chain disruptions, it is worth noting that its integration makes the supply chain and, consequently, its modelling more complex. This requires, therefore, the development of efficient, fast, and relevant resolution approaches.

1.5 . Research questions

The research project is motivated by real-world applications. The investigation of these fundamental problems reveals a specific need for suppliers

and customers relative to the management of inventories and goods transportation and delivery. They are seeking new approaches to work together to reduce costs while guaranteeing high service levels and preventing stock-outs as much as possible. This has revitalised the classical IRP under new constraints and brought the critical question of its resolution to the fore.

The present research is also driven by gaps in the literature regarding modelling and solving complex vehicle routing problems with shared inventory management in supply chains. Yet, it is well known that large inventory routing problems are complicated to solve to optimality and are notoriously challenging for exact methods that push researchers to design suitable solutions.

In this thesis, we are advocating for a new management policy promoting multi-sourcing options to mitigate shortages. This policy combines regular shipment from manufacturer to customers, inventory sharing through LT among customers, and the use of product substitutions whenever it is possible. We assume emergency supplies from the CW are significantly extended and expensive, which makes relevant investigating substitution whenever possible along with LT whenever demand exceeds the available stock at the level of each customer.

Therefore, both the case studies and the literature analysis open promising lines of research. Particularly, the main objective of this thesis can be stated as follows:

Modelling and solving complex vehicle routing problems with integrated management of shared inventories in supply chains.

Four main research questions can be derived from this research objective:

1. How to mathematically model inventory sharing and substitutions constraints within inventory routing problem formulations under different customer demand and supply chain settings?
2. How to design suitable solving approaches integrating as much as possible the knowledge built up thanks to the mathematical modelling?
3. How to quantify the benefits of sharing inventories between suppliers and customers, and what advantages does product substitution bring to the different players?
4. Under which organisation conditions inventory sharing is the most beneficial to the involved players?

To address these questions, an appropriate methodology is described hereafter.

1.6 . Research methodology and dissertation organisation

1.6.1 . Research methodology

We establish a global review of the literature in parallel with a deep understanding of the business cases. This step helps us depict an explicit characterisation and classification of the different problems and encoded configurations. We then develop mathematical formulations to model the various forms of IRP decision problems with transshipment and substitutions. After that, we design appropriate and original solving approaches. The performance of these solving methods is compared to the best existing benchmark. Finally, different experiments are conducted to evaluate the benefits of the proposed approaches for managing shared inventory.

Based on the above-described practical cases and thorough literature reviews, we identify four constrained and complicated configurations of real-world supply chains so that the benefits of promoting inventory sharing and product substitution are highlighted. As the availability of both finished products and RTIs is a vital input for an effective performance appraisal, we mainly investigate the distribution of fashion items, perishable products, and spare parts for finished goods and the management of RTIs in the automotive and food industries in closed-loop supply chains. The configurations under study are as follows:

- For finished products:
 - We consider a two-level supply chain in which a company manufactures a set of products and sells them through its point of sale network (we refer to as customers). A deterministic multi-product multi-vehicle inventory routing problem with LT in which transshipment-related decisions are integrated into the design of routes is studied (cf. Chapter 2).
 - We conserve the same configuration, and along with LT, substitutions of products are used to sidestep shortages at the customer level in stochastic contexts. The two modes of managing the pickup and delivery of shared inventory are examined (cf. Chapter 3 and 4).
 - We investigate the same configuration but this time with decentralised decision-making. A deterministic multi-product multi-vehicle Vehicle Routing Problem (VRP) with LT and Inventory Management (VRP-TIM) is solved. The problem is modelled as a 1- leader - n -followers Stackelberg game, and the LT-related decisions are integrated into the design of vehicle routing (cf. Chapter 5)

- For RTIs:
 - We consider a two-level-closed loop supply chain consisting of a set of suppliers and customers. A deterministic, multi-supplier, multi-customer inventory routing problem with pickup and delivery of multi and shared RTI is modelled and solved (cf. Chapter 6).
 - We consider a stochastic, multi-supplier, multi-hub, multi-customer inventory routing problem with pickup and delivery of multi and shared containers in the PI's related logistics services. Also, inventory sharing between the hubs, including empty RTIs and finished products, is considered (cf. Chapter 7).

Figure 1.1 provides a global view of the different configurations under study.

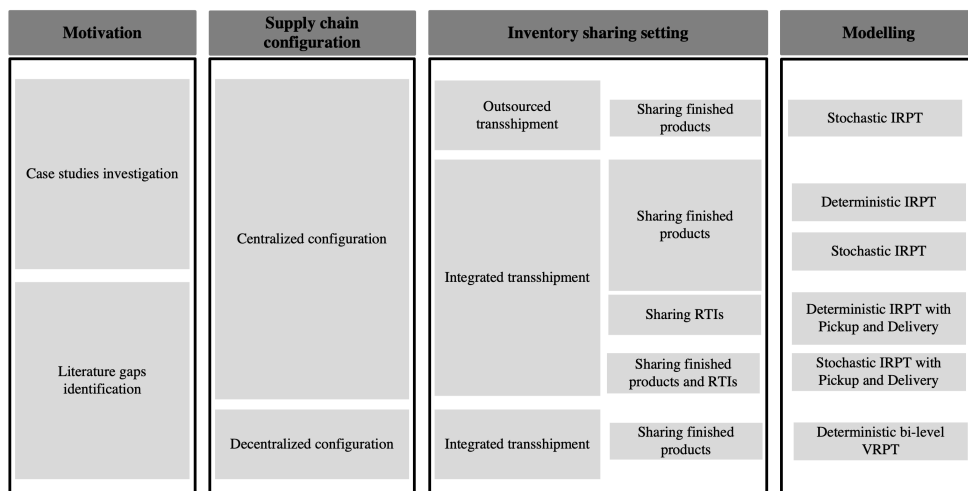


Figure 1.1: Overview of the configurations under study

Regarding the resolution approaches, we investigate first exact methods and meta-heuristics as they are widely used for classical IRP. Literature reviews revealed that few algorithms could be applied to the problems studied in this thesis. Original solution approaches- called matheuristics - based on the hybridisation of mathematical modelling, sophisticated metaheuristics and deep reinforcement learning techniques (RL) are then developed.

We propose an original matheuristic to solve a deterministic IRPT that integrates mathematical modelling strengthened with relevant derived valid inequalities and hybridises two sophisticated metaheuristics: Genetic Algorithm (GA) and Simulated Annealing (SA). Also, we used Sample Average Approximation and GA coupled with mathematical modelling and RL

to solve a stochastic IRPT with integrated/outsourced LT and substitutions (cf. Chapters 3 and 4).

The competitive performance of the matheuristic coupled with RL encourages us to use it to solve the models developed in Chapters 5, 6 and 7. Regarding managerial insights, the results highlight the benefits of inventory sharing and substitution on the overall performance of the supply chain: cost reduction (up to 50%) and improved service levels (up to 43%). Results also suggest insights of interest to professionals willing to develop new decision support models for the most efficient management of finished products and RTIs inventory.

Further details are given in Chapters 2, 3, 4, 5, 6 and 7. Contributions related to these questions are published in international journals and conferences. Figure 1.2 provides an overview of the main contributions that had been carried out in this doctoral thesis. It's worth highlighting that each scientific paper has a specific methodology presented in each article.

1.6.2 . Dissertation organisation

Figure 1.3 gives a synoptic view of how the thesis is structured based on literature gaps and case studies described in the previous sections. Chapter 2 presents the model and resolution approach developed to solve a deterministic multi-product multi-vehicle IRPT in which transshipment-related decisions are integrated into the design of routes. Computational results highlighted the benefits of LT on the overall supply chain. Also, a sensitivity analysis is conducted, revealing the extent to which inventory sharing among different points of sale can be cost-effective by the savings it brings to reduce lost sales and inventory holding.

Based on the supply chain configuration and modelling given in Chapter 2, Chapter 3 provides the model with and resolution approach developed to solve a static and stochastic multi-product multi-vehicle spare parts IRP with substitution and LT, and in which transshipment-related decisions are integrated into the design of routes. Computational experiments highlight the benefits of promoting LT and substitution on the supply chain performance. Results also suggest insights that interest professionals willing to develop new decision support models for the most efficient management of such items.

Chapter 4 extends the work presented in Chapter 3 by investigating a dynamic and stochastic version of the problem, considering that LT is outsourced, and by combining the matheuristic, proposed in Chapter 2, with RL techniques to enhance its performance. Results confirm the efficiency of the proposed algorithm and highlight the benefits of both transshipment, and substitutions on the supply chain's overall performance.

The competitive performance of the resolution approach described in Chapter 5 encourages us to use it to solve a deterministic multi-product multi-vehicle VRP with LT and inventory management under a decentralised setting of the

General Objective: Modelling and solving complex vehicle routing problems with integrated management of shared inventories in supply Chains					
Main objective	Research type	Research method	Data generation	Model analysis	Publication status
Article 1 A metaheuristic for solving inventory sharing problems.	Quantitative research	Mathematical modelling and metaheuristics	Academic benchmark and random generation	Descriptive quantitative analysis of model/algorithm performance	Full paper published in Computers and Operations Research (Q1, IF=4)
Article 2 Spare parts inventory routing problem with transshipment and substitutions under stochastic demands.	Quantitative research	Mathematical modelling	Academic benchmark and random generation	Descriptive quantitative analysis of model/algorithm performance	Full paper published in Applied Mathematical Modelling (Q1, IF=5)
Article 3 Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning	Quantitative research	Mathematical modelling, metaheuristics, and reinforcement learning	Academic benchmark and random generation	Descriptive quantitative analysis of model/algorithm performance	Full paper published in International Journal of Production Research (Q1, IF=8)
Article 4 Bi-level programming for modelling inventory sharing in decentralized supply chains	Quantitative research	Mathematical modelling, metaheuristics, Game theory, and reinforcement learning	Academic benchmark and random generation	Descriptive quantitative analysis of model/algorithm performance	Ready to be submitted to Journal of Intelligent Transportation (Q1, IF=4)
Article 5 Solving Returnable Transport Items Problems using an Artificial Immune System based Algorithm enhanced with Deep Reinforcement Learning	Quantitative research	Mathematical modelling, metaheuristics, and reinforcement learning	Random generation	Descriptive quantitative analysis of model/algorithm performance	Full paper published in Sustainability (Q1, IF=3)
Article 6 Integrating Forward and Reverse Network in the Interconnected Logistics Services in the Physical Internet	Quantitative research	Mathematical modelling, metaheuristics, and reinforcement learning	Random generation	Descriptive quantitative analysis of model/algorithm performance	Ready to be submitted to International Journal of Production Research (Q1, IF=8)

Figure 1.2: Summary of the research methodology

configuration presented in the previous chapters. The gap analysis shows that the proposed algorithm performs relatively well and that inventory sharing allows the network to improve its service level.

After highlighting the benefits of promoting sharing inventory of finished products, sharing empty RTIs inventory is examined in closed-loop supply chain configurations. In Chapter 6, a deterministic, multi-supplier, multi-customer inventory routing problem with pickup and delivery of multi and shared RTIs is modelled and solved. From a managerial point of view, the results stress that sharing RTIs allows economies of scale and cost reduction at the level of all involved parties. From a managerial point of view, the results emphasise that this new approach will enable economies of scale and cost reduction at the level of all involved parties. Also, a sensitivity analysis highlights the benefits and limits of the proposed model compared to classical management strategies.

To bring into play the advantages of sharing finished products and their reusable packaging, we exploit the PI paradigm characteristics in terms of high integration, flexibility and openness. Chapter 7 is built on models, configurations and solving methods examined in previous chapters. It focuses on the inventory routing model with pickup and delivery of reusable PI-containers, which is new to the literature. Also, deliveries from suppliers to the PI-hubs and from PI-hubs to the customers are considered to be performed within time windows. Furthermore, the positive impact of the PI model on logistics efficiency compared to the classical model is highlighted.

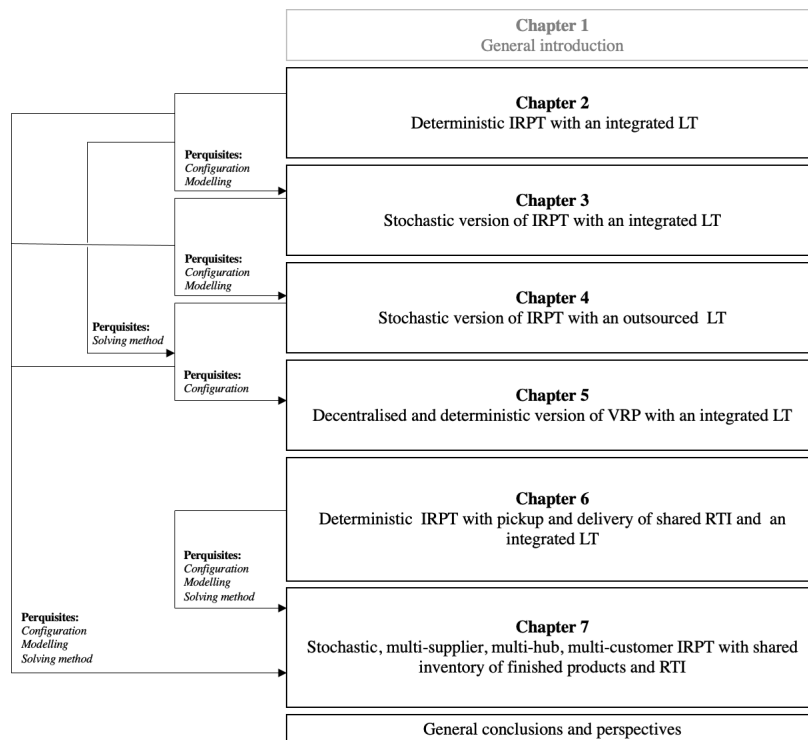


Figure 1.3: Dissertation organisation

The list of publications associated with this thesis and on which this manuscript is built is as follows:

- Chapter 2:
Achamrah, F.E., Riane, F., Di Martinelly, C. and Limbourg, S. (2022). A matheuristic for solving inventory sharing problems. *Computers & Operations Research*.
- Chapter 3:
Achamrah, F.E., Riane, F. and Limbourg, S. (2022). Spare parts inventory routing problem with transshipment and substitutions under stochastic demands. *Applied Mathematical Modelling*.
- Chapter 4:
Achamrah, F.E., Riane, F. and Limbourg, S. (2021). Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning. *International Journal of Production Research*.
- Chapter 5:
Achamrah F.E., Riane F., and Aghezzaf E.H. (2022). Bi-level program-

ming for modelling inventory sharing in decentralized supply chains. Transportation Research Procedia.

- Chapter 6:
Achamrah F.E., Riane F., Bouras A., and Sahin E. (2020). Collaboration Mechanism for Shared Returnable Transport Items in Closed Loop Supply Chains', 9th International Conference on Operations Research and Enterprise Systems, ICORES 2020, VALLETA, Malta.
Achamrah F.E, Riane F., Sahin E., and Limbourg, S. (2022). An Artificial-Immune-System-Based Algorithm Enhanced with Deep Reinforcement Learning for Solving Returnable Transport Item Problems. Sustainability.
- Chapter 7:
Achamrah F.E, Riane F., and Aghezzaf E.H. (2022). Integrating Forward and Reverse Network in the Interconnected Logistics Services in the Physical Internet. Under review.

2 - Modelling and solving a deterministic multi-product multi-vehicle IRP-T

Full Paper published in *Computers & Operations Research Journal*, ISSN 03050548, 1873765X, Impact Factor: 4, Quartile: Q1. <https://doi.org/10.1016/j.cor.2021.105605>

Achamrah, F.E., Riane, F., Di Martinelly, C. and Limbourg, S. (2022). A heuristic for solving inventory sharing problems. *Computers & Operations Research*. Volume 138, 2022, 105605, ISSN 0305-0548.

A Matheuristic for Solving Inventory Sharing Problems

Fatima Ezzahra Achamrah ^{a,b}, Fouad Riane ^{a,b}, Christine Di Martinelly ^c,
Sabine Limbourg ^d

^aComplex Systems and Interactions, Ecole Centrale of Casablanca; ^bLaboratoire Genie Industriel, CentraleSupélec, Paris Saclay University; ^c IESEG School of Management, Univ. Lille, ^d HEC-University of Liege (ULiege)

Abstract: In this paper, we addressed a two-level supply chain in which a company manufactures products and sells them through its point of sale network. The problem of concern is a multi-product, multi-vehicle inventory sharing routing problem. We formulated it as a mixed-integer linear programming problem. We also designed a two-phase matheuristic that integrates mathematical modelling strengthened with relevant derived valid inequalities and hybridisation of sophisticated metaheuristics: Genetic Algorithm and Simulated Annealing. We tested 660 best known instances designed for the single and multi-vehicle inventory routing problem, and the results showed that the matheuristic outperformed the best known algorithms regarding computational times. Extra experiments were conducted on a set of data specially designed for the multi-product multi-vehicle inventory sharing routing problem. Computational results highlighted the benefits of sharing inventory on the overall supply chain. Finally, a sensitivity analysis was conducted, which revealed the extent to which inventory sharing among different points of sale can be cost-effective by the savings it brings to reduce lost sales and inventory holding.

Keywords: Shared Inventory, Transshipment, Inventory Routing Problem, Matheuristic

2.1 . Introduction

In this paper, we focus on an inventory sharing problem that arises in a fashion distribution company. The company manufactures products and allocates them to its customers (i.e., Point Of Sales POS). Customer demand varies, and stock imbalance can regularly happen: some customers may be overstocking items while others may be running out of them. The IHL Group, a global research and advisory firm specialising in technologies for the retail and hospitality industries, revealed that \$1.1 trillion worldwide total lost revenue opportunities is annually recorded due to inventory distortion (IHL, 2015). Furthermore, acquiring trucks to handle last-mile distribution

is becoming more complex as cities are getting more congested (Viu-Roig and Alvarez-Palau, 2020). To ensure that the customers have enough items so that the company can optimise its revenue, it can be in its best interests to allow inventory sharing between its customers to balance inventories and enhance service levels. Moreover, the same fleet of vehicles can be used to ship products from its central warehouse (CW) to the customers and between customers when needed. Furthermore, the fact that the same vehicles are used to perform regular shipment and transshipment leads to better use of each vehicle's carrying capacity and implicitly reduces the carbon footprint. This paper is positioned in the overlapping area of the two well-known problems: the classical inventory routing problem (IRP) and the transshipment problem in which the same fleet of vehicles is used to make regular shipments from CW to reallocate stocks between customers. The related literature is now briefly examined.

2.1.1 . IRP and transshipment

The classical IRP refers to a combination of the vehicle routing and inventory management problem in which a supplier has to deliver products to several geographically dispersed customers (i.e., retailers, POS), subject to side constraints (Coelho et al., 2014b). IRP arises in a vendor-managed inventory (VMI) setting in which a supplier manages the inventory replenishment of its customers so that the supplier saves on distribution cost while the customers save on inventory management cost (Campbell et al., 1998). The customers will engage in this relationship with the supplier once a high service level is guaranteed. Some applications can be found in Absi et al. (2013); Karakostas et al. (2020); Karakostas et al. (2020); Wu et al. (2021); Gunawan et al. (2019) and Li et al. (2013).

However, regarding highly variable demand, IRP may lead to higher inventory costs and a bullwhip effect. To mitigate this issue, sharing inventory between the different locations belonging to the same distribution network level can result in a considerable cost reduction (Paterson et al., 2011). This type of inventory sharing is commonly referred to as LT (Grahovac and Chakravarty, 2001). In their literature review on LT, Paterson et al. (2011) divided the research into proactive and reactive transshipment. Proactive transshipment is conducted periodically to re-balance the entire system's stock levels and aims to reduce potential shortages (Dehghani et al., 2021). Reactive transshipment reacts to a scenario where one of the locations faces a shortage while others have residual stock in hand (Cavagnini et al., 2018). This type of transshipment is suitably conducted in environments with a relatively low transportation cost compared to the holding and shortage cost. Reactive transshipment can be encountered either in centralised systems where a single decision-maker works to increase the overall profit of the entire system or in decentralised systems where each decision-maker op-

erates to maximise profits. Most studies on centralised systems suppose that the transshipment lead times are lower than those of the regular shipment (Axsäter, 2003; Banerjee et al., 2003; Burton and Banerjee, 2005; Shao et al., 2011; Mangal and Chandna, 2010; Herer and Tzur, 2001). The authors showed that an LT policy is preferable to a non-LT policy if the advantages of preventing shortages exceed the increased delivery cost resulting from transshipment. When significant LT times are considered, the benefits of risk pooling are substantial only when the demand is highly variable (Grahovac and Chakravarty, 2001; Tagaras and Vlachos, 2002; Wong et al., 2005; Kutanoglu and Mahajan, 2009). Moreover, when considering stochastic demands, transshipment can show more flexibility and better results regarding lost sales reduction (Cavagnini et al., 2018; Abouee-Mehrzi et al., 2015).

It is in the latter context that Coelho et al. (2012b) introduced formally the concept of IRP with transshipment (IRP-T). Under such a policy, products may be shipped to a customer, either directly from the supplier or from other customers. The total cost to be minimised includes inventory holding cost at the supplier and the customers, routing and transshipment costs. The authors proposed a single-vehicle IRP-T under Order-Up to level (OU) and Maximum Level (ML) policies. The authors assumed that shortages are restricted, and transshipment is performed by a carrier's vehicles, not by the supplier's vehicles. Transshipment decisions are therefore not considered when optimising the routing decisions between the manufacturer and customers. The authors used an Adaptive Large Neighbourhood Search heuristic (ALNS) to tackle large-scale instances. Lefever et al. (2018) extended the work of Coelho et al. (2012b). To strengthen the mathematical formulation, the authors proposed a set of valid inequalities for IRP-T based on the existing valid inequalities for the IRP, bounds, reformulation and variable eliminations on the linear relaxation of the problem of concern. Mirzapour Al-e-hashem and Rekik (2014) explored an IRP-T with a heterogeneous fleet of trucks. They considered many-to-one supply chain networks comprising one assembly plant and a set of suppliers, each of which supplies only one product type. The authors assumed that a rental truck company is responsible for shipping products from suppliers to the plant.

The MILP is solved by employing a branch-and-bound method. Azadeh et al. (2017) proposed an MILP for IRP-T in the presence of a single perishable product. Under an ML policy, a supplier decides on the quantities to deliver to each customer over a finite planning horizon. The authors used the Genetic Algorithm (GA) to solve the problem at hand and the Tagushi approach for adjusting its parameters. Turan et al. (2017) addressed an IRP-T under stochastic demand. They investigated a supply chain comprising a central warehouse and retail stores. Transshipment may occur between retail stores within a time window to re-balance the inventories. To solve the prob-

lem, they used a Variable Neighbourhood Search algorithm. [Timajchi et al. \(2019\)](#) studied an IRP-T for hazardous and deteriorating pharmaceutical items in a healthcare network. Transshipment is allowed between hospitals to sidestep any shortage. They proposed a bi-objective MILP. The first objective aimed to minimise the total cost, including ordering, transportation, delivery, pickup, shortage and inventory holding costs. The second objective function minimised the maximum accident loss during distribution among all periods. The authors solved the problem using GA. Finally, [Peres et al. \(2017\)](#) modelled a multi-product inventory routing problem for a proactive transshipment between a factory and a set of distribution centres. In their paper, multi-customer routes are considered an alternative to direct shipping. Indeed, during a planned route, instead of shipping products to each single centre, a vehicle can be filled with the current demand of a centre and the future demand of other centres. The pre-delivered demand is stored temporarily in this specific centre. The authors use a Randomised Variable Neighbourhood Descent to solve the problem. Table 2.1 characterises the previous papers regarding criterion integration of multi-product (MP), multi-vehicle (MV), Shortage (SH), either Backorder (BO) or Lost sales (LS), transshipment mode (TM) if LT is outsourced (OS) or integrated into the routing (IT). Finally, resolution approaches (RA) use: MILP, metaheuristic (MT) or a combination of the latter (matheuristic: MTH). From Table 2.1, a few papers on IRP-T studied a multi-product multi-vehicle version and considered lost sales due to the shortages as a measure of the quality of the service and customer satisfaction. Also, transshipment-related decisions were ignored in designing vehicle routing as the transshipment is always assumed to be subcontracted or performed by another carrier. Indeed, to simplify the optimisation problem, the authors only determine the nodes and periods where transshipment may occur and manage inventories so that transshipment can be performed. This paper thus aims to fill this gap.

Table 2.1: Overview of the related work on IRP-T

References	MP	MV	SH		TM		RA		
			BO	LS	OS	IT	MILP	MT	MTH
Dehghani et al. (2021)	✓			✓	✓		✓		
Coelho et al. (2012b)					✓		✓	✓	
Lefever et al. (2018)					✓		✓		
Mirzapour Al-e-hashem and Rekik (2014)	✓	✓			✓		✓	✓	
Azadeh et al. (2017)					✓		✓	✓	
Turan et al. (2017)	✓	✓			✓		✓	✓	
Timajchi et al. (2019)			✓	✓	✓		✓		
Peres et al. (2017)	✓	✓			✓		✓	✓	
This paper	✓	✓		✓		✓	✓		✓

2.1.2 . Pickup and delivery

Integrating LT in the design of routes that do regular shipment overlaps another classical problem: the pickup-and-delivery problem (PDP). PDP refers to the collection and distribution of one or several products to and

from a location (Berbeglia et al., 2007). According to Berbeglia et al. (2007) and Battarra et al. (2014), the PDP can be classified into three variants – Many-to-many (M-M), one-to-many-to-one (1-M-1) and one-to-one (1-1) – based on the pickup and delivery structures. In an M-M PDP, multiple pickups and delivery nodes exist (Xu et al., 2017). 1-M-1 PDP refers to the deliveries from one depot to many customers, and pickups are performed to send back products to the depot (Wassan et al., 2017). Finally, a 1-1 PDP refers to pickup and delivery requests in which each pickup location is assigned to one delivery location (Qu and Bard, 2012). Applications of the PDP combined with IRP can be mostly found in maritime transportation. For reviews on maritime transportation, the reader is referred to Christiansen et al. (2013) and Christiansen and Fagerholt (2014). On-road transportation, Iassinovskaia et al. (2017) studied 1-M-1 IRP with simultaneous pickups and deliveries of the returnable transport item in a two-level supply chain. Archetti et al. (2018) explored an IRP with pickup and delivery (IRP-PD) in which a commodity must be picked up from pickup customers and delivered to delivery customers over a given planning horizon with a single capacitated vehicle. Archetti et al. (2020) extended this work to the multiple vehicle case. In both papers, it is assumed that role of pickup or delivery customers remains unchanged during all the time horizons. Our study parallels that of van Anholt et al. (2016) as it also considers two PDP structures in their model: the 1-M-1 structure, which accounts for product shipments from a depot to automated machines and back to a depot, and the M-M structure, which refers to a commodity shipment among machines. However, in the IRP-PD under consideration, routes are mainly constructed for pickup and delivery. At the same time, in this paper, transshipment is considered an additional measure to sidestep shortages, and transshipment movements, depending on the costs, could be permitted on a link (i, j) even if they were not to be used in IRP without transshipment. Moreover, van Anholt et al. (2016) focused on one product while considering the multi-product case. Also, in their paper, machines are assumed to be visited either for a pickup or delivery operation, which is not the case in our paper. Each POS can be a pickup node, delivery node, or both in each period. Finally, the return flows of products to the CW were unexplored. We believe that we are unaware of a contribution to multi-product IRP-T that includes these aspects mentioned above.

2.1.3 . Scientific contribution and organisation of the paper

This paper contributes to the literature in three main dimensions. First, it studies a multi-product multi-vehicle IRP-T in which transshipment-related decisions are integrated into the design of routes. Each vehicle is allowed to deliver regular demand of each product: from a CW to customers and transshipped products from a pickup node to a delivery node. Second, it proposes

an original matheuristic capable of handling the combinatorial complexity of the problem, which integrates mathematical modelling strengthened with relevant derived valid inequalities and hybridisation of two sophisticated meta-heuristics. The performance of the suggested matheuristic is extensively tested on benchmark sets of the literature for single and multi-vehicle IRP, and results are compared to the best-known algorithms for similar problems. Finally, extra experiments are conducted to explore inventory sharing benefits and draw managerial insights regarding multi-product multi-vehicle IRP-T.

The remainder of the paper is organised as follows. Sections 2.2 and 2.3 provide the description and mathematical formulation of the problem or concern, respectively. Section 2.4 describes the approach proposed to tackle the problem complexity and derive good solutions in reasonable computational times. In Section 2.5, computational results are presented to evaluate the approach's accuracy and performance and get insight into the benefits of promoting inventory sharing. Conclusions and perspectives are drawn in Section 2.6.

2.2 . Problem setting

In this paper, we investigate a fashion distribution company that manufactures a set of products and allocates them via a CW to its customers according to a pre-established policy based on the previous sales of similar products. In most cases, some customers have over-stock (and therefore incur high holding cost), while others are out of stock (and therefore incur shortage cost: loss of sales). The company aims to increase its revenues and service level and avoid any loss of profit due to the non-availability of products at the right places and moments. Therefore, it is in the company's interest to pool the stock of its customers and agree to manage all the inventory collectively by entrusting operations optimisation to CW. We formulate the problem as an MILP based on a VMI system. The optimisation is centralised, i.e., the CW optimises the total cost by deciding on the stock levels of each customer and the routes to be constructed. Transshipment occurs to sidestep possible stock-outs. Moreover, lost sales happen when a shortage occurs, as we assume that backorders are restricted. Lost sales are used to measure the service level at the customer locations.

The customer is located at distinct geographic locations, and we assume there is no competition for the demand. Each customer orders from the manufacturer's CW to meet its demand. The demand for items is assumed to be deterministic and variable over the planning horizon. Based on a VMI system and an ML policy, the manufacturer is responsible for delivering the products to the different POSs using a homogeneous fleet of vehicles with a limited capacity. The CW attempts to meet the customer's demand

while minimising its cost over a finite planning horizon. We consider that transshipment is possible between customer locations (see Figure 2.1 for an illustrated example). We also assume that the manufacturer's emergency shipments are significantly longer and more expensive than transshipment. The sequence of operations performed at a node in each time period is the following: first, the product is delivered, then the demand is satisfied, and finally, the inventory level is calculated. Finally, it is assumed that the quantities delivered in period t from the CW cannot be transshipped in period t .

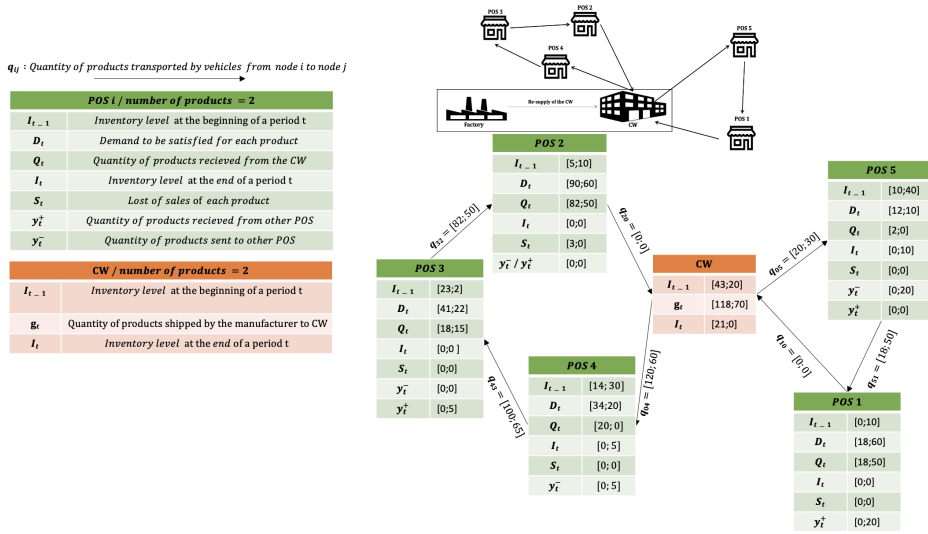


Figure 2.1: A numerical example of the IRP-T with 2 products.

The objective is to minimise the total cost while increasing the availability of items by allowing LT between the customer and coordinating the supply and demand through shared inventory management and vehicle routing optimisation. Over a finite planning horizon, solving the IRP-T determines for each period the optimal distribution routes from the CW to the set of customers and how to build the inventories such that the customer can meet the demands of their customers.

2.3 . Mathematical formulation

IRP-T is defined on a graph $G = (\mathcal{N}_0, \mathcal{A})$, where $\mathcal{N}_0 = \{0, \dots, n\}$ is the vertex set and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}_0, i \neq j\}$ is the edge set. Let $\mathcal{N} = \{1, \dots, n\}$ be the set of n customers and 0 be the vertex representing the manufacturer's CW. At the end of each period, both the CW and customers incur an inventory holding cost h_{pi} ($i \in \mathcal{N}$) per product $p \in \mathcal{P} = \{0, \dots, m\}$. Each CW and customer has a maximum inventory

holding capacity C_i ($i \in \mathcal{N}_0$). The length of the planning horizon is T with time period $t \in \mathcal{H} = \{1, \dots, T\}$. At the beginning of the planning horizon, the current inventory level I_{pi0} , which is expressed in terms of the stock keeping unit (SKU) for each $i \in \mathcal{N}_0$, is known. In addition, the manufacturer receives information on the demand of each product D_{pit} that each customer has to satisfy for each period t . $v \in V = \{1, \dots, k\}$ is the available set of k homogeneous vehicles. Let us define \mathcal{Q} (in equivalent SKU) as the vehicle capacity and α as the associated transportation cost per km. Let d_{ij} (expressed in km) be the associated distance for each arc $(i, j) \in \mathcal{A}$, and b_{ij} be the additional transportation unit cost associated to transshipping products from a customer i to a customer j . The total quantity delivered to a customer in a given period guarantees that the storing capacity is not exceeded at the end of the period. f_{pi} is the lost sales cost associated with product p at location i . The quantity of an item p planned to be shipped by the manufacturer to the CW at period t is g_{pt} .

All these notations are summarised in Table 4.1.

Table 2.2: Notation summary

\mathcal{H}	Planning horizon indexed by t
V	Set of vehicles indexed by v
\mathcal{P}	Set of products indexed p
Parameters	
α	Transportation cost per km
h_{pi}	Unit inventory holding cost per period for product p of node, $i \in \mathcal{N}_0$
C_i	Maximum inventory capacity at node $i \in \mathcal{N}_0$
I_{pi0}	Inventory level of a product p at location $i \in \mathcal{N}_0$ at the beginning of the planning horizon
D_{pit}	Demand for product p to be satisfied at period t at node $i \in \mathcal{N}$
\mathcal{Q}	Vehicle capacity
$d_{i,j}$	Distance associated to arc $(i, j) \in \mathcal{A}$
g_{pt}	Quantity of product p shipped by the manufacturer to the CW in period t
b_{ij}	Unit cost induced by transshipping products from node $i \in \mathcal{N}$ to node $j \in \mathcal{N}$
f_{pi}	Unit loss of sales cost associated with the product p at the level of the customer $i \in \mathcal{N}$
Variables	
I_{pit}	Inventory level of product p at node $i \in \mathcal{N}_0$ at the end of period t
Q_{pit}	Quantity of product p directly shipped from the CW to the node $i \in \mathcal{N}$ in a period t
q_{pijvt}	Quantity of product p transported from the node $i \in \mathcal{N}_0$ to the node $j \in \mathcal{N}_0$ by vehicle v in a period t . It includes regular shipment from CW and transshipment between customers
y_{pijvt}	Quantity of product p transshipped from the node $i \in \mathcal{N}$ to the node $j \in \mathcal{N}$, by vehicle v to sidestep the shortage of product p in a period t
S_{pit}	Lost sales quantity of product p the node $i \in \mathcal{N}$ in a period t
x_{ijvt}	= 1 if the arc $(i, j) \in \mathcal{A}$ is visited by a vehicle v in period t ; 0 otherwise
u_{vt}	= 1 if the vehicle v is used in period t ; 0 otherwise
z_{ivt}	Equal to 1 if and only if a node $i \in \mathcal{N}_0$ is visited by vehicle v in period t (defined for valid inequalities)

The objective is to minimise the cost of inventory, routing and transshipment while meeting the demand for each customer and avoiding shortages that may induce loss of sales over the planning horizon as much as possible. The formulation of the IRP-T can be written as:

$$\begin{aligned}
\min \left[\sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}_0} \sum_{p \in \mathcal{P}} h_{pi} I_{pit} + \sum_{t \in \mathcal{H}} \sum_{v \in V} \sum_{i, j \in \mathcal{N}_0} \alpha d_{ij} x_{ijvt} \right. \\
\left. + \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{H}} \sum_{v \in V} \sum_{i, j \in \mathcal{N}, i \neq j} b_{pj} y_{pijvt} + \sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}} \sum_{p \in \mathcal{P}} f_{pi} S_{pit} \right] \quad (2.1)
\end{aligned}$$

Subject to:

$$I_{pit} = I_{pit-1} + Q_{pit} - D_{pit} + S_{pit} + \sum_{v \in V} \sum_{\substack{j \in \mathcal{N} \\ i \neq j}} (y_{pijvt} - y_{pjivt})$$

$$\forall p \in \mathcal{P}, i \in \mathcal{N}, t > 0 \in \mathcal{H} \quad (2.2)$$

$$I_{p0t} = I_{p0t-1} - \sum_{i \in \mathcal{N}} Q_{pit} + g_{pt} \forall p \in \mathcal{P}, t > 0 \in \mathcal{H} \quad (2.3)$$

$$Q_{pjvt} + \sum_{v \in V} \sum_{\substack{i \in \mathcal{N} \\ i \neq j}} (y_{pijvt} - y_{pjivt}) = \sum_{v \in V} \sum_{\substack{i \in \mathcal{N}_0 \\ i \neq j}} (q_{pijvt} - q_{pjivt})$$

$$\forall p \in \mathcal{P}, j \in \mathcal{N}, t \in \mathcal{H} \quad (2.4)$$

$$\sum_{p \in \mathcal{P}} q_{pi0vt} = 0 \quad \forall i \in \mathcal{N}_0, v \in V, t \in \mathcal{H} \quad (2.5)$$

$$\sum_{p \in \mathcal{P}} I_{pit} \leq C_i \quad \forall i \in \mathcal{N}_0, t \in \mathcal{H} \quad (2.6)$$

$$\sum_{p \in \mathcal{P}} q_{pijvt} \leq \mathcal{Q}u_{vt} \quad \forall i, j \in \mathcal{N}_0, v \in V, t \in \mathcal{H} \quad (2.7)$$

$$\sum_{v \in V} \sum_{j \neq i \in \mathcal{N}_0} y_{pijvt} \leq I_{pit-1} \quad \forall p \in \mathcal{P}, i \in \mathcal{N}, t > 1 \in \mathcal{H} \quad (2.8)$$

$$\sum_{i \neq j \in \mathcal{N}_0} x_{ijvt} = \sum_{i \neq j \in \mathcal{N}_0} x_{jivt} \quad \forall j \in \mathcal{N}, v \in V, t \in \mathcal{H} \quad (2.9)$$

$$\sum_{i \neq j \in \mathcal{N}_0} \sum_{v \in V} x_{ijvt} \leq 1 \quad \forall j \in \mathcal{N}, t \in \mathcal{H} \quad (2.10)$$

$$\sum_{j \in \mathcal{N}} x_{0jvt} = u_{vt} \quad \forall v \in V, t \in \mathcal{H} \quad (2.11)$$

$$\sum_{v \in V} u_{vt} \leq k \quad \forall t \in \mathcal{H} \quad (2.12)$$

$$q_{pijvt} \leq \mathcal{Q} x_{ijvt} \quad \forall i, j \in \mathcal{N}, v \in V, t \in \mathcal{H} \quad (2.13)$$

$$Q_{pit}, I_{pit}, y_{pijvt}, q_{pijvt}, S_{pit} \in \mathbb{N} \quad \forall p \in \mathcal{P}, i, j \in \mathcal{N}_0, v \in V, t \in \mathcal{H} \quad (2.14)$$

$$x_{ijvt}, u_{vt} \in \{0, 1\} \quad \text{for all } i, j \in \mathcal{N}_0, v \in V, t \in \mathcal{H} \quad (2.15)$$

The objective function (7.2) minimises the total cost. The first sum corresponds to the inventory cost at both the CW and customer locations. The

second sum stands for transportation cost. The third sum captures the transshipment cost. The fourth sum is for the lost sales cost at different customer locations. Constraints (7.4) ensure the satisfaction of demand D_{pit} expressed at customer i for product p in period t . This is possible thanks to the quantity of product p available at customer i , which is equal to the inventory level at the beginning of the period computed at the end of the period (I_{pit-1}) plus the quantity of the product p delivered either directly from the CW, Q_{pit} , or transshipped from another customer j , minus the total quantity of product p transshipped from i to different customer. A surplus constitutes the inventory level I_{pit} of product p at the end of period t . Otherwise, a shortage will be registered, leading to a lost sale S_{pit} . Constraints (7.5) express the conservation conditions of the inventory at the CW over successive periods. The conditions consider the quantities of product p delivered to the CW from the manufacturer, and those shipped to the different customers. Constraints (7.6) express the flow conservation conditions at a customer j . Constraints (7.7) state that the vehicles must be empty at the end of each period when returned to the CW. Constraints (7.8) guarantee that the inventory levels at different locations i do not exceed the maximum inventory capacity. Constraints (7.9) state that the vehicle's capacity is not exceeded for each period and arc. Constraints (7.11) state that the quantity transshipped from customer i at a period t does not exceed the initial inventory level available in this period. Constraints (7.13) stipulate that if a vehicle v enters (visits) the customer j in period t , it must leave it in the same period. Constraints (7.15) ensure that, at most, a vehicle v visits a customer once per period. Constraints (7.17) ensure that only vehicles carrying products leave the CW. Constraints (7.18) stipulate that the sum of vehicles used in a time period t respect the total number of available vehicles. Constraints (7.19) ensure that product p will only be transported from a node i to a node j by vehicle v if the arc (i, j) is being used by the vehicle v during the same period t . Constraints (7.21)–(2.15) define the non-negativity and binary conditions.

Valid inequalities can be added to the formulations of the IRP problem to improve the quality of the root node lower bound (Coelho and Laporte, 2014; Lefever et al., 2018). Several researchers have developed valid inequalities for IRP formulations. Below, we present different classes of such inequalities that we modified to make them compatible with our IRP-T model.

We start with constraints referred to as logical inequalities commonly used for the multi-vehicle IRP. Constraints (2.16) impose the condition that if a supplier (node 0, in our case CW) is the predecessor of a node i on vehicle v 's route, then the same vehicle must visit i , as follows (Archetti et al., 2007):

$$x_{0ivt} \leq z_{ivt} \quad \forall i \in \mathcal{N}, v \in V, t \in \mathcal{H} \quad (2.16)$$

Constraints (2.17), which are also referred to as logical inequalities, impose the condition that if a node i is the predecessor of a node j on the vehicle v path, then the same vehicle must visit j , as follows (Archetti et al., 2007):

$$x_{ijvt} \leq z_{jvt} \quad \forall i, j \in \mathcal{N}, v \in V, t \in \mathcal{H} \quad (2.17)$$

Constraints (2.18) insert the supplier (CW) in the route of vehicle v if it visits any customer in the given period, as follows (Archetti et al., 2007):

$$z_{ivt} \leq z_{0vt} \quad \forall i \in \mathcal{N}, v \in V, t \in \mathcal{H} \quad (2.18)$$

Constraints (2.19) deal with symmetry breaking and state that vehicle v cannot leave the depot if vehicle $v - 1$ is not used, as follows (Coelho and Laporte, 2014):

$$z_{0vt} \leq z_{0v-1t} \quad \forall i \in \mathcal{N}, v \in V \setminus \{1\}, t \in \mathcal{H} \quad (2.19)$$

Constraints (2.20), which were developed based on the valid inequalities of Coelho and Laporte (2014) for IRP and extended for the single-vehicle single-product IRP-T by Lefever et al. (2018), state that if the inventory level of a customer j is insufficient to fulfil the demand of product p , then a visit must take place, as follows:

$$\sum_{t=t_1}^{t_2} \sum_{v \in V} z_{ivt} + \frac{\sum_{t=t_1}^{t_2} \sum_{j \in \mathcal{N}} \sum_{v \in V} y_{pjivt}}{\sum_{t=t_1}^{t_2} D_{pit}} \geq \frac{\sum_{t=t_1}^{t_2} D_{pit} - I_{pit-1}}{\sum_{t=t_1}^{t_2} D_{pit}} \quad (2.20)$$

$\forall i \in \mathcal{N}, p \in \mathcal{P}, t_1, t_2 \in \mathcal{H}, t_2 \geq t_1.$

For each product $p \in \mathcal{P}$, the inventory level at t_1 is sufficient to cover the demand as long as the left hand side of Constraints (2.20) is less or equal to zero. Otherwise, either the customer should be supplied by the CW or transshipment from other customer should be allowed to meet the demand.

Avella et al. (2018) proposed disjoint path inequalities for a single product IRP considering the demands and inventory levels from the periods $t-1$ and $t+1$. Once violated, cuts are generated based on the arc disjunction of a given route S to eliminate edges from the sub routes of S . According to the authors, these inequalities help to enhance the overall performance of the algorithm. The formulation of these inequalities is modified to use them in our model, as follows:

$$\begin{aligned} & \sum_{i,j \in \mathcal{A}} \mu_{ij} x_{ijvt} \geq \left[\sum_{j \in S_0 \cup S_1} \sum_{p \in \mathcal{P}} q_{pijt} + \sum_{j \in S_2} \sum_{p \in \mathcal{P}} (q_{pijt} - I_{pj,t+1}) \right. \\ & \left. + \sum_{j \in S_3} \sum_{p \in \mathcal{P}} (q_{pijt} - I_{pj,t}) + \sum_{j \in S_4} \sum_{p \in \mathcal{P}} \left(I_{pj,t} - \left(C_j - \sum_{t-1 < t' < t+1} D_{pj,t'} \right) \right) \right] \quad \forall i \in \mathcal{N}, v \in V, t \in \mathcal{H} \end{aligned} \quad (2.21)$$

where $(S_0, S_1, S_2, S_3, S_4)$ are a partition of $S \subseteq N$ and $\mu_{ij} = \min(Q - b_i, b_j)$ whose values lead to a disjoint route inequality, such that $b_i = Q$ if $i \in S_0$, $b_i = C_i$ if $i \in S_1$, $b_i = \sum_{t < t' < t+1} D_{pit}$ if $i \in S_2$, $b_i = D_{pit}$ if $i \in S_3$, $b_i = D_{pit-1}$ if $i \in S_4$ and $b_i = 0$ for $i \in \mathcal{N}_0 \setminus S$. Referring to [Avella et al. \(2018\)](#), the disjoint route inequality holds for every route R starting and ending at the CW. Indeed, regardless of the number of products to ship, it suffices to consider the case where $\sum_{i,j \in R} \mu_{ij} < \mathcal{Q}$ and thus assume that $\sum_{i,j \in R} \mu_{ij} \geq \sum_{i \in V(R) \cap S} b_i$ to prove that this claim holds for any route; and thus, it follows that it is a disjoint route inequality.

The mathematical formulation is solved using the branch-and-cut method as follows. As some valid inequalities are of exponential size, they are not added in the beginning but separated on-the-fly. If a valid inequality is violated, it is dynamically added at the current branch-and-cut node, and the program is then re-optimised. This is handled by first enumerating all valid inequalities and then checking them one by one in order. The process is repeated until a feasible or dominated solution is found or no more cuts need to be made. As for the branching strategy, it is set to default which means that no automatic priority order will be generated.

2.4 . Resolution Approach

IRP and its variant are first solved using exact methods, inter alia, branch-and-bound ([Mirzapour Al-e-hashem and Rezik, 2014](#)), branch-and-cut ([Archetti et al., 2007](#)), column generation algorithms [Michel and Vanderbeck \(2012\)](#) and dynamical constraint-generating algorithm ([Lefever et al., 2018](#)). Nevertheless, IRP and IRP-T are extensions of the Vehicle Routing Problem, including inventory decisions. ([Laporte, 2009](#)) proved that VRP problems are NP-hard. Thus, solving such problems for medium-to-large size scales requires using approximation algorithms. Accordingly, significant attention and research have been devoted to developing effective approximate algorithms that can provide nearly optimal solutions to large-scale problems, including the IRP and its variants.

A matheuristic algorithm with two optimisation stages is proposed in this study (see Figure 2.2). Since the route design is the most complex part of the model, an initial solution for route decisions is built during the construction phase. This is made possible by solving a relaxed model version strengthened by valid inequalities (RMILP). If the obtained solution is optimal, the algorithm stops. The objective is to get a better solution without going through the improvement phase and thus reduce further the computational time. Otherwise, this solution is used to construct clusters of customers. For each built cluster, the non-relaxed mathematical model is then solved to set vehicles' routing. In an iterative process, a hybridisation of GA and SA is

used to further improve the best solution found by generating a set of neighbours of the route decisions. An MILP in which fixed routes (noted FMILP) is used to assess each neighbour's quality and feasibility and determine the other integer variables, such as the inventory level and quantity directly delivered from CW and shared between customers. The entire process stops when the annealing temperature reaches a final temperature, a time limit is reached, or no solution improvement is noted. We now extensively describe these two phases.

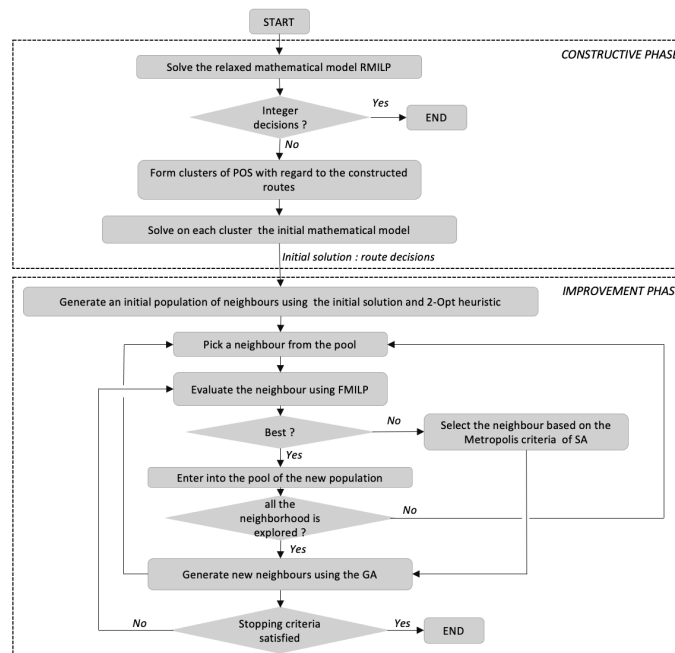


Figure 2.2: Decomposition scheme of the matheuristic.

2.4.1 . Constructive phase

The objective is to decompose the problem into sub-problems that will be solved to obtain route decisions. The decomposition allows us to reduce the size of the problem and, consequently, the computational times. This is done in two steps. First, we consider a relaxed version of the model noted RMILP comprising Constraints (7.4)–(7.19), (2.15) and (2.16)–(2.21), and in which all decision variables, except routing's variable, are considered continuous. Solving RMILP allows us to obtain a solution to the problem at hand. If the solution is not optimal, we retain only the constructed routes that allow us to build a set of sub-graphs or clusters for each period (as shown in Figure 2.3-a). Each sub-graph or cluster contains a subset of customers (Figure 2.3-b). Unlike classical clustering, which is based on minimising distances in and between clusters and in which the number of clusters is often equal to the number of available vehicles (Nananukul, 2013), this clustering is more

efficient as its construction is based not only on distances but also on the constraints related to the inventory levels, quantity shipped and shared, and in which the number of clusters varies accordingly. Next, non-negativity Constraints (7.21) related to each decision variable are added to the RMILP model as lazy constraints. Then the related MILP is solved for each constructed sub-graph (as shown in Figure 2.3-c). Moreover, the algorithm is executed until a time limit is reached or a feasible solution is found. The objective is to generate a good initial solution within a small amount of time as it is claimed that the initial solution affects the efficiency of an algorithm to find a solution near the optimum. Finally, we retain the solution related to the route decisions, which will be further improved in the second stage. The pseudo-code of this procedure is provided in Algorithm 1.

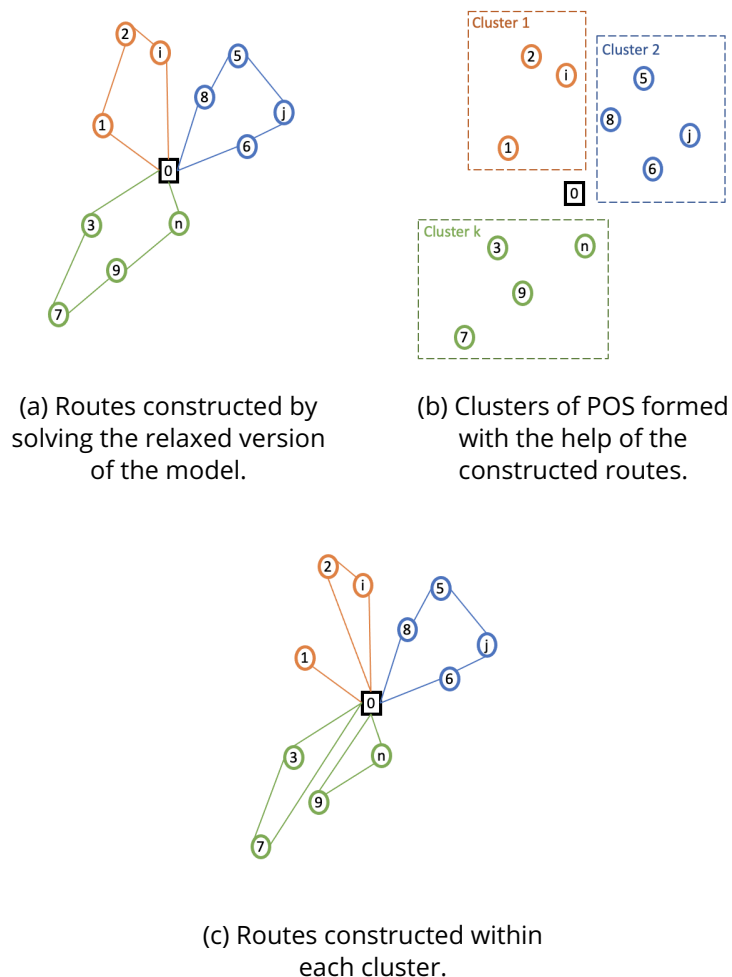


Figure 2.3: Implementing the construction phase of the matheuristic in a given period.

Algorithm 1 : Constructive phase

Input : A given instance
Output : Routing decisions
Solve RMILP
Use route decisions to build a set of sub-graphs/clusters
Add lazy constraints
Solve concurrently the MILP on each sub-graph to obtain routes decision

2.4.2 . Improvement phase

In the improvement stage, the routing decisions obtained in the constructive phase are iteratively improved using the metaheuristics SA and GA and an MILP in which routes are fixed (noted FMILP). It has be noted that, as routes are under continuous adjustment thanks to GA and SA operators, new neighbourhoods are explored, allowing better flexibility and an enhanced space search for feasible solutions. In this phase, an initial population of neighbours of the routing decisions obtained in the first stage (noted X) is randomly generated using the 2-Opt permutation technique. Without loss of generality, each generated neighbour is noted X' . Also, each neighbour X' is picked, and, if feasible, its fitness $F(X')$ is computed through the resolution of the related FMILP. If the new fitness exceeds the old one, X is given a new value, X' . Even if the new solution is worse, it can still be accepted using the Metropolis acceptance criteria of SA. Iteratively, the algorithm generates the neighbourhood of the current solution using the genetic operators of the GA and evaluates it using FMILP. Furthermore, to enhance the algorithm's local search ability, worse solutions accepted according to the Metropolis criteria are also used to generate new neighbourhoods. We now extensively describe these steps.

2.4.2.1 . Genetic Algorithm

The improvement process starts with an initial set of random solutions called population. Each individual in the population is referred to as a chromosome X , representing a routing decision to the problem at hand. The fitness of each solution is measured during each generation of a chromosome, and solutions are selected for cloning, crossover and mutation based on their fitness (computed using objective function values).

Chromosomes encoding : in this paper, each chromosome X is represented using a one-dimensional array of integer values, representing the nodes (customers) to be visited (see Figure 2.4). Each chromosome is further partitioned into several sub-sequences, each representing a constructed route assigned to a given vehicle and in a given period (see Figure 4.3). This partition allows us to verify the routing constraints. It helps to easily check

whether a customer with a non-zero demand is missing on the routes or if it is served several times in the same period. It also helps GA operators to perform better and quickly. For instance, if we swap two nodes, this partition ensures that they either belong to the same route or two different routes performed during the same period.

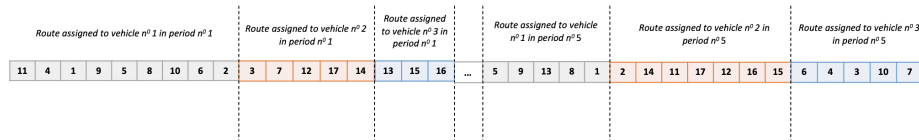


Figure 2.4: Representation of a partitioned chromosome used to solve the problem at hand.

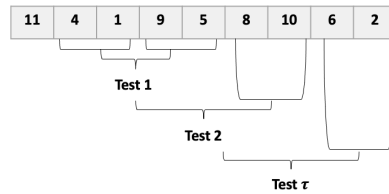


Figure 2.5: A chromosome part representing a constructed route for a given period and vehicle.

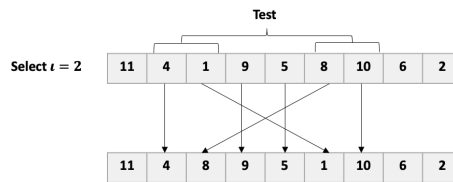
Finally, each chromosome is re-converted to binary variables x_{ijvt} (while also considering the values of x_{0ivt} and x_{i0vt}) to solve the FMILP and then determine its related fitness.

Initial population generation : A 2-Opt heuristic is an algorithm based on the conditional permutation of nodes. In other words, two nodes in a current tour are selected and swapped if it reduces the total distance. The process can be repeated several times until the tour is optimised. The computational complexity of this method is $O(n^2)$. A variant of this heuristic proposed by [Sabba and Chikhi \(2012\)](#) shows promising results regarding the quality of the solutions and the computational times (a few seconds). It encourages us to use this variant to generate an initial population for our algorithm. The heuristic is thus applied to a randomly selected tour, and the resulting chromosome is added to the population. As shown in Figure 2.6, this algorithm comprises of selecting a position ι in a current tour, then the permutation of nodes is authorised between the two segments « $C_\iota C_{\iota+1}$ » if it allows reducing total cost. This verification is performed with all segments which follow the segment « $C_\iota C_{\iota+1}$ ». If the permutation allows a cost reduction, then only the second node of the first segment and the first node of the second segment can be swapped, while the intermediate nodes keep

the same position. Furthermore, the enhanced 2-Opt also relies on **inter-route moves**. That is, swapping two nodes that belong to two different tours performed in the same period is also permitted.



(a) Test phase



(b) Permutation of nodes

Figure 2.6: Overview of the enhanced 2-Opt proposed by Sabba and Chikhi (2012)

Fitness function The fitness function of a chromosome X is calculated from the objective function $OF(X)$ as follows:

$$F(X) = \frac{1}{OF(X)} \quad (2.22)$$

Genetic Operators : in this algorithm, the following operators are used:

- Cloning operator, which preserves the best solutions found so far. That is, chromosomes are reproduced in each generation, and this is done by selecting a member of the population according to its fitness and making a copy. This guarantees that the best members of the current population will be present in the next. Finding the trade-off between the algorithm's performance and speed led us to choose the best 30% of chromosomes in the current population to be copied to the next generation.
- Parent selection operator, which uses a binary tournament selection process that starts with the creation of two pairs of chromosomes. Every pair comprises two chromosomes that are randomly selected from the current population. The two best chromosomes for each

pair are chosen for crossover operations. This process results in two children, each counted in the new population.

- Crossover operator is important to mate the pairs of chromosomes to produce their offspring. Herein, a double-point crossover is implemented in which the crossover points are randomly selected, and the nodes between are swapped to produce two children. A crossover is performed based on a probability P_C and the partition adopted for each chromosome. Figure 2.7 shows the detailed procedure.
- Mutation operator is the second operator used for exploring new neighbours. The aim is to produce random alterations in different chromosomes. This paper adopts reversal mutation, which has been shown to be efficient (Zhang et al., 2010). Two nodes are randomly selected (based on the representation in Figure 4.3), and the nodes in between are sorted in the reverse order. A mutation process is performed using a probability P_M .

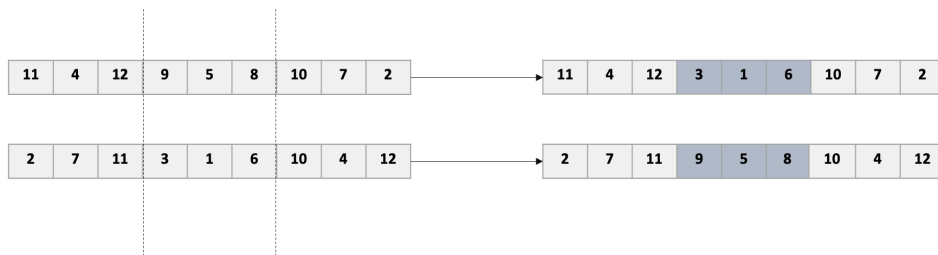


Figure 2.7: Crossover operation.

2.4.2.2 . Simulated Annealing

GA is preferred to solve problems efficiently by generation through its systemic operators, which allow for the improvement and variety of solutions. However, GA has inherently weak local search ability and converges prematurely. On the other hand, SA is a local search metaheuristic that provides a mechanism for avoiding local optima by accepting hill-climbing movements. The SA algorithm provides a different (or a new neighbour of the current) possible solution by changing the current state at each virtual annealing temperature by a criterion previously specified. On the basis that the Metropolis criterion is met, the new state will then be accepted. Until convergence, this procedure will be iterated.

Practically, the algorithm employs this characteristic as follows. As depicted in Figure 2.8, it begins by generating, aided by the 2-Opt heuristic, an initial population of the routing decision X determined in the con-

structive phase. A given chromosome X' is picked from the initial generation, its related fitness is computed, and if it outperforms the old one (i.e., $F(X') > F(X)$), the current routing X is given a new value, which is X' . Even if the new solution is worse (i.e., $F(X') < F(X)$), it can still be accepted using the Metropolis acceptance criteria of SA to enhance the local search ability of the algorithm. That is, a chromosome X' is accepted with the probability $P >$ a random number generated between 0 and 1 (Equation 2.23 is used to compute P). Otherwise, X' is rejected. For each annealing temperature T_{SA} , this process is repeated a given maximum number of iterations, $iter$, equal to the size of each generated population. When $iter$ is reached, c used as a variable for incrementing is reset to 1; the annealing temperature is updated using the function βT_{SA} , where β is the cooling ratio ($0 < \beta < 1$). New populations of neighbours X' are generated using the GA operators. Moreover, worse solutions accepted according to the Metropolis criteria are also used to generate new neighbours. In this paper, duplicated offspring are rejected alongside neighbours generating infeasible solutions. $Iter$ is updated regularly to take the value of the new population size. The entire process stops when the annealing temperature T_{SA} reaches a final temperature T_f , or a time limit is reached, or no improvement of the solution is noted. Therefore, the best $F_{best}(X)$ and the related decision variables are obtained.

$$P = e^{-\frac{(OF(X') - OF(X))}{T_{SA}}} \quad (2.23)$$

2.5 . Computational experiments and discussion

In this section, we assess the effectiveness and efficiency of the proposed model and the matheuristic approach. We first describe the experiments and benchmark instances used to evaluate the efficiency of the matheuristic on a single-product IRP and compare it to benchmark algorithms. The matheuristic is also tested on specially designed instances for multi-product IRP-T so that the benefits of promoting inventory sharing among different POSs can be highlighted. We then explain how we fine-tune the metaheuristic. To investigate interactions between the parameters, a sensitivity analysis is performed.

2.5.1 . Experiment design and instances

First, to test the effectiveness and validate the proposed model, we perform experiments on a set of well-known benchmarks of 660 instances developed for a single and multi-vehicle IRP. The set comprises 480 small instances and 180 large ones. 160 small instances proposed by Archetti et al. (2007)

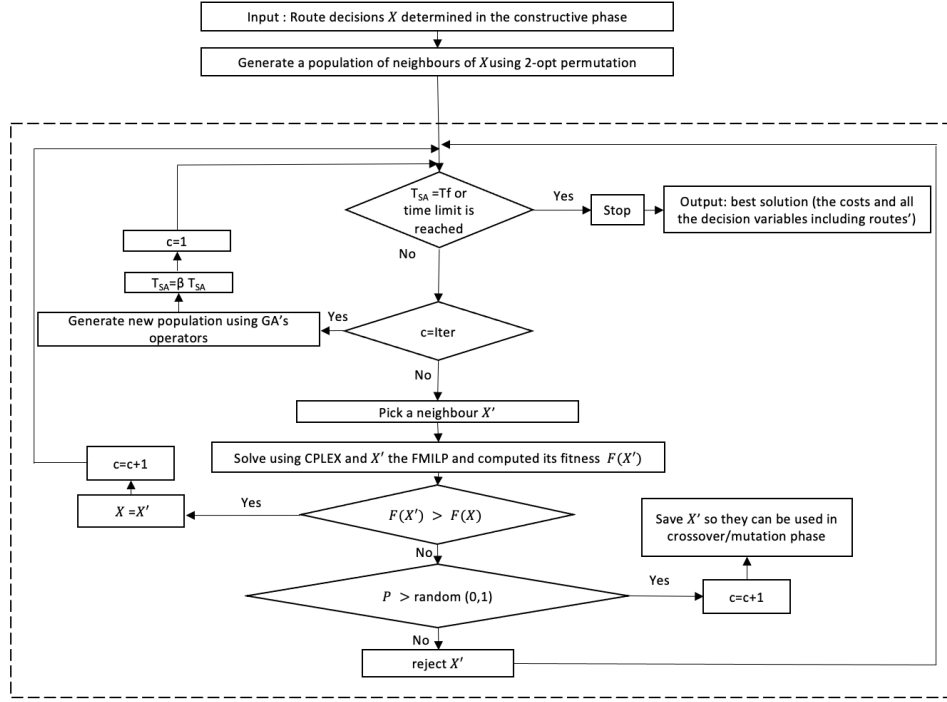


Figure 2.8: Flow diagram of the improvement phase.

and 60 large ones proposed by Archetti et al. (2012) for a single-vehicle IRP. In addition, 320 small instances and 120 large ones adapted for multi-vehicle case ($k = \{2, 3\}$) by Coelho and Laporte (2013b). For each case, small instances comprise 100 three-period instances with 5 to 50 customers and 60 six-period instances with a number of customers varying between 5 and 30. Large instances comprise of 6-periods and 50 to 200 customer instances. Each configuration can be further divided into two subgroups according to the distributions of their unit inventory costs, which are $[0.01, 0.05]$ at the customers and equal to 0.03 at the CW (low cost) and $[0.1, 0.5]$ at the customers and equal to 0.3 at the CW (high cost). In addition, the transportation cost between each pair of vertices (i, j) is computed by rounding the Euclidean distance, and the lost sale cost is equal to 200 times the unit inventory cost. As these instances were developed for the single and multi-vehicle IRP, we adopt the proposed model by considering the LT's and lost sales-related decision variables:

$$y_{pijvt} = 0 \quad \forall p \in \mathcal{P}, i, j \in \mathcal{N}_0, v \in V, t > 0 \in \mathcal{H} \quad (2.24)$$

$$S_{pit} = 0 \quad \forall p \in \mathcal{P}, i \in \mathcal{N}_0, t > 0 \in \mathcal{H} \quad (2.25)$$

In the instances mentioned above, we compare the results of our proposed matheuristic to the results obtained using the best known exact algorithm

branch-and-cut (B-C) of [Coelho and Laporte \(2013b, 2014\)](#) with a time limit up to 86400 s and the recently improved branch-and-cut (I-B-C) developed by [Guimarães et al. \(2020\)](#). The reader, if interested, is referred to this paper for further details on I-B-C. The I-B-C is tested on 800 instances generated for the single and multi-vehicle IRP and compared to the-state-of-the-art algorithms: the B-C of [Avella et al. \(2018\)](#) and [Adulyasak et al. \(2014\)](#), which were tested only on the small instances, and the B-C of [Coelho and Laporte \(2013b\)](#), and the heuristic of [Archetti et al. \(2012\)](#) which, were tested on small and large instances generated by [Archetti et al. \(2007\)](#) and [Archetti et al. \(2012\)](#). With a time limit of 7200 s, the I-B-C outperforms the existing algorithms regarding the quality of solutions and run time ([Guimarães et al., 2020](#)). The authors highlight that their algorithm reaches 704 best-known solutions, 108 exclusive ones on small instances of [Archetti et al. \(2007\)](#) and 70 new best solutions for large instances of [Archetti et al. \(2012\)](#). Therefore, instead of comparing our matheuristic with other state-of-the-art algorithms, we compare it with I-B-C. The computational platforms for the B-C and I-B-C are Xeon CPU 2.66 GHz and Xeon CPU E5-2630 v2 2.6 GHz, respectively. All instances and detailed solutions for the exact methods are retrieved and available from: <https://www.leandro-coelho.com/instances/inventory-routing/>.

For a fair comparison between approaches, hardware benchmarking is used to compare the speed of the algorithms. The reported CPU of the matheuristic is thus recalculated to align the computational time regarding the performance of computers used in [Coelho and Laporte \(2013b, 2014\)](#), and [Guimarães et al. \(2020\)](#). For each algorithm, CPU and scaled CPU (S_CPU) are reported. Further information on the CPU speed of both computers can be found on: www.cpubenchmark.net.

Second, we consider a set of 50 randomly generated instances to evaluate the multi-product IRP-T and highlight the benefit of LT on the supply chain's overall performance. The small and large instances were generated following the same standards as in [Archetti et al. \(2007\)](#), [Archetti et al. \(2012\)](#) and [Coelho and Laporte \(2013a\)](#). The number of products varies between 1 and 30. As for transshipment unit cost, as in [Coelho et al. \(2012b\)](#), we choose $b_{ij} = 0.01\alpha d_{ij}$.

2.5.2 . Parameters setting of the matheuristic

Experiments have shown that computational times become very important for some parameter values, even for small-sized instances. To tune the parameters of the GA and the SA, we use the Irace package ([López-Ibáñez et al., 2016](#)). The Irace package implemented in R uses the Iterated Race method: a generalisation of the Iterated F-race method for the automatic configuration of optimisation algorithms. The method comprises three steps: (1) sampling new configurations according to a particular distribution, (2) selecting the best configurations from the newly sampled ones through racing,

and (3) updating the sampling distribution to bias the sampling towards the best configurations. These three steps are repeated until a termination criterion is met (López-Ibáñez et al., 2016). We use a set of training instances representing the problem (20 instances with 5, 10, 15... 50 customers each) to find the best algorithm configuration for the GA and SA (see Table 2.3). The selected algorithm configuration can then be used to solve new instances of the same problem.

Table 2.3: Parameters tuning using Irace package

Parameters	Range	Chosen values
Initial temperature T_i	[90 ; 95]	92
Final temperature T_f	[3.5 ; 5.1]	4.2
Cooling ratio β	[0.94 ; 0.97]	0.96
Crossover probability P_C	[0.81 ; 0.85]	0.84
Mutation probability P_M	[0.35 ; 0.43]	0.37
Population size	[150 ; 200]	180
Number of iteration $iter$	[150 ; 200]	180

2.5.3 . Computational experiments

The computational experiments were conducted on a personal computer (MacBook Pro, macOS Cataline, CPU 3.3 GHz Quad-Core Intel Core i7, 8 GB of RAM). The problem was solved using the branch-and-cut solver of CPLEX 12.9 (academic version) and Python 3.7. We performed 25 independent runs for each instance under a time limit of 2 hours (7200 s) and recorded the results and run-time average.

2.5.3.1 . Single and multi-vehicle IRP

This section presents the results of the experiments performed on the small and large-scale datasets. Table 2.4 summarises the results for small instances. The first four columns give the number of the vehicle (K), the number of period T , the distribution of inventory cost IC (low/high), and the number of instances NI generated for each T and IC . For each exact method, we give the number of the best solution found (NBFS), the gap related to the lower bound (GLB) in (%) computed using B-C and I-B-C and finally, the computational time (CPU) in the second. For the matheuristic, statistics are provided regarding the average of the best solutions. We thus note \overline{GLB} and \overline{CPU} the average of the gap computed using the lower bound of the exact methods and the average CPU time, respectively. Finally, $\overline{S_CPU}$ represents the average of the scaled CPU.

From Table 2.4, the proposed matheuristic can find optimal solutions for all 160 small-scale instances for single-vehicle IRP within a minimal amount of time compared to B-C and I-B-C. Moreover, for the multi-vehicle cases under consideration, on average, both I-B-C and matheuristic outperform B-C in terms of the number of the optimal solution found and run time. Finally, for the 320 instances, the matheuristic outperforms I-B-C, the current

best resolution approach regarding computational times (CPU and S_CPU), highlighting its performance and efficiency.

Table 2.4: Results for small-scale benchmark instances obtained using B-C, I-B-C and the matheuristic

Instances				B-C			I-B-C			Matheuristic			
K	T	IC	NI	NBSF	GLB	CPU	NBSF	GLB	CPU	NBSF	\overline{GLB}	\overline{CPU}	$\overline{S_CPU}$
1	3	Low	50	50	0.0	10	50	0.0	29	50	0.0	1	7
	3	High	50	50	0.0	10	50	0.0	26	50	0.0	2	9
	6	Low	30	30	0.0	38	30	0.0	134	30	0.0	3	14
	6	High	30	30	0.0	29	30	0.0	94	30	0.0	3	16
2	3	Low	50	49	0.0	1539	50	0.0	457	50	0.0	12	61
	3	High	50	50	0.0	672	49	0.0	455	50	0.0	41	204
	6	Low	30	27	0.3	10942	28	0.4	1716	28	0.1	39	194
	6	High	30	28	0.1	7234	28	0.1	1516	28	0.1	35	173
3	3	Low	50	42	1.3	8790	43	0.7	1896	43	0.3	58	291
	3	High	50	42	0.5	8772	42	0.3	1686	41	0.4	77	383
	6	Low	30	15	5.0	27046	16	5.0	4426	16	0.3	65	326
	6	High	30	14	2.6	25385	17	2.0	4161	17	1.9	49	243

Detailed results on large instances are presented in Tables of Appendix A. The tables provide the worst solution (WS), the best solution (BS), the average of the best solutions found (\overline{BFS}), and the average \overline{GLB} computed regarding the LB of each exact algorithm, and the average of the scaled CPU ($\overline{S_CPU}$) regarding hardware performances (for a fair comparison, the actual CPU was limited that its corresponding \overline{CPU} would be comparable to the limit time of I-B-C: 7200 s). The tables also provide the contribution of the constructive phase (CPC in %) in improving the quality of the final solution. Also, for each $\overline{S_CPU}$, tables report the computational time needed in the CP and improvement phase (IP).

We refer to each instance using the following notation: [number of products] P [number of POS] N [number of vehicles] K [number of periods] T, e.g., 2P5N2K5T refers to the instance where two items are distributed over a set of 5 POS, transported by two vehicles over a planning horizon of 5 days. We retain the notation in the rest of the paper. Finally, Table 2.5 provides a summary of comparison of the results regarding the average \overline{UB} , \overline{GLB} and \overline{CPU} (which represents the $\overline{S_CPU}$ for the matheuristic).

Table 2.5: Summary of comparison of results for large benchmark instances obtained using B-C, I-B-C and the matheuristic

Instances				B-C		I-B-C		Matheuristic	
K	T	IC	NI	\overline{UB}	$\overline{CPU}(s)$	\overline{UB}	$\overline{CPU}(s)$	\overline{UB}	$\overline{CPU}(s)$
1	6	Low	30	19911	69058	16535	5798	16879	2315
	6	High	30	72078	59960	64431	5326	65681	2692
2	6	Low	30	31061	86400	18804	7200	19689	2732
	6	High	30	60487	86400	66886	7200	69679	3148
3	6	Low	30	42348	86400	22235	7200	23062	3401
	6	High	30	84955	86400	70505	7200	73382	3981

From Table 2.5, the proposed matheuristic can find solutions within the

least amount of computational time compared to B-C and I-B-C due to its exploration capability regarding the searching procedures. Indeed, the matheuristic can find solutions with an average \overline{CPU} of 3045 s against 6654 s for I-B-C and 79103 s for B-C. As for solution quality, we can see that both I-B-C and the matheuristic outperform B-C and that I-B-C has the lowest \overline{UB} compared to our matheuristic. Moreover, both approaches can find good exclusive solutions. On the other hand, we can observe that, unlike small instances, the matheuristic does not scale quite well with large instances. In this case, there is an exponential increase in search space size, which requires an increase in the GA population size to explore the space further. In addition, we can see from Tables in Appendix A that the constructive phase is the most time-consuming component. Indeed, in the constructive phase, if the solution is not optimal, the model is solved optimally on each constructed cluster, which could be time-consuming depending on the size of instances. To tackle this problem, one can argue that we could solve the model without Constraints (7.21). However, adding them as lazy constraints significantly helps generate excellent initial solutions (clearly highlighted in Tables of Appendix A), which enhances the efficiency of the matheuristic to find a good final solution.

2.5.3.2 . Results for IRP-T on small and large instances

For each instance, we consider two scenarios in which the IRP-T is compared to a model that does not consider LT between customers, IRP. Tables 2.6 and 2.7 summarise the results obtained for small and large-scale instances, with some products varying between 1 and 5 for small instances and 10 and 30 for large ones. Tables 2.6 and 2.7 provide the breakdown of the total cost (TC), transportation (TR), Inventory (INV), lost sales (LS) and transshipment (TRA). They also report the improvement in service level (SL) and cost-saving (CS) regarding the reduction of the total cost and, finally, the un-scaled \overline{CPU} , as there is no need to use $\overline{S_CPU}$. The measure of the attained \mathcal{S} is computed for each instance as follows:

$$\mathcal{S}(\%) = \left(1 - \frac{\sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}} \sum_{p \in \mathcal{P}} S_{pit}}{\sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}} \sum_{p \in \mathcal{P}} D_{pit}}\right) 100$$

The saving CS is computed as follows:

$$CS(\%) = \left(\frac{TC_{IRP} - TC_{IRP-T}}{TC_{IRP}}\right) 100$$

and the improvement in the service level SL as:

$$SL(\%) = \left(\frac{\mathcal{S}_{IRP} - \mathcal{S}_{IRP-T}}{\mathcal{S}_{IRP}}\right) 100$$

Table 2.6: Computational results for multi-product IRP-T on small scale instances

Instances	Model	Cost (\$)					Improvement		\overline{CPU} (s)
		TR	INV	LS	TRA	TC*	CS (%)	SL(%)	
1P5N8K5T	IRP	970	940	1170	0	3080	16	24	1
	IRP-T	1080	912	460	121	2573			0
2P5N8K5T	IRP	1062	1056	1160	0	3279	25	44	12
	IRP-T	1154	892	270	147	2462			10
3P5N8K5T	IRP	1175	1109	1670	0	3954	31	45	6
	IRP-T	1259	814	420	220	2714			9
4P5N8K5T	IRP	1239	1209	1940	0	4388	39	47	2
	IRP-T	1376	809	200	282	2667			4
5P5N8K5T	IRP	1379	1320	2090	0	4789	31	40	9
	IRP-T	1422	805	760	336	3324			7
1P10N8K5T	IRP	2110	1490	2370	0	5970	21	38	1
	IRP-T	2349	1045	950	350	4694			3
2P10N8K5T	IRP	2215	1591	5060	0	8866	27	37	9
	IRP-T	2521	1307	2190	470	6487			7
3P10N8K5T	IRP	2435	1782	1470	0	5687	16	45	13
	IRP-T	2565	1281	340	599	4785			5
4P10N8K5T	IRP	2579	1901	3520	0	8000	21	36	8
	IRP-T	2768	1265	1620	643	6296			12
5P10N8K5T	IRP	2762	2129	2500	0	7391	21	41	11
	IRP-T	2939	1181	850	838	5809			12
1P15N8K5T	IRP	3527	2912	3940	0	10379	13	33	12
	IRP-T	3925	2425	2130	563	9043			7
2P15N8K5T	IRP	3771	3543	3330	0	10645	26	48	7
	IRP-T	4090	2646	400	690	7826			3
3P15N8K5T	IRP	4245	4165	3340	0	11751	20	34	10
	IRP-T	4436	2513	1640	795	9385			13
4P15N8K5T	IRP	4765	4563	2900	0	12227	26	41	8
	IRP-T	4736	2509	970	835	9050			1
5P15N8K5T	IRP	5314	4986	4350	0	14650	35	44	7
	IRP-T	5201	2378	940	955	9474			2
1P20N8K5T	IRP	6242	4122	3650	0	14014	15	37	13
	IRP-T	6947	2798	1560	540	11845			6
2P20N8K5T	IRP	6714	4398	4660	0	15773	14	32	13
	IRP-T	7548	2646	2530	849	13573			6
3P20N8K5T	IRP	7479	4635	4880	0	16994	22	45	10
	IRP-T	7808	3466	970	994	13238			5
4P20N8K5T	IRP	8126	5212	3760	0	17098	18	40	6
	IRP-T	8398	3304	1350	1012	14064			14
5P20N8K5T	IRP	8539	5546	4550	0	18635	22	43	4
	IRP-T	8691	3246	1320	1240	14496			1

* All obtained solutions are optimal

Table 2.7: Results for multi-product IRP-T on large scale instances

Instances	Model	Cost (\$)					Improvement		\overline{CPU} (s)
		TR	INV	LS	TRA	TC	CS (%)	SL (%)	
10P50N8K5T	IRP	29806	32895	53010	0	115711*	46	32	485
	IRP-T	30964	26458	4499	620	62540*			287
15P50N8K5T	IRP	52304	33060	65286	0	150650	19	33	513
	IRP-T	49626	26618	44489	1107	121839			753
20P50N8K5T	IRP	41633	43329	90540	0	175502	38	38	512
	IRP-T	43554	26030	37972	1541	109096*			282
25P50N8K5T	IRP	34860	33189	184778	0	252827	50	41	562
	IRP-T	38077	29937	55488	2685	126187			507
30P50N8K5T	IRP	45724	37389	137590	0	220703	42	43	431
	IRP-T	53325	19583	51118	3496	127523			411
10P100N8K5T	IRP	111543	43445	178128	0	333115	40	38	600
	IRP-T	113309	34722	49504	1962	199496			747
15P100N8K5T	IRP	122592	47667	52029	0	222289	14	36	566
	IRP-T	123608	39160	25541	3270	191579			437
20P100N8K5T	IRP	132149	58444	66273	0	256866	19	41	567
	IRP-T	133852	49178	22018	2560	207608			311
25P100N8K5T	IRP	184874	43945	107364	0	336184	18	37	882
	IRP-T	192418	38681	40589	3651	275339			581
30P100N8K5T	IRP	204793	46082	92821	0	343696	15	37	615
	IRP-T	205111	37601	42658	7491	292861			742
10P150N8K5T	IRP	204998	69472	169692	0	444162	26	37	414
	IRP-T	206209	53397	63908	3982	327495			1126
15P150N8K5T	IRP	321715	56038	129293	0	507047	9	37	711
	IRP-T	342635	40783	71763	5466	460647			303
20P150N8K5T	IRP	494281	65768	185640	0	745690	22	44	835
	IRP-T	481456	58675	35928	6116	582174			467
25P150N8K5T	IRP	506862	79054	73959	0	659875	9	43	1161
	IRP-T	517423	60444	18591	6887	603345			611
30P150N8K5T	IRP	624238	80723	77893	0	782854	10	40	228
	IRP-T	628745	42873	29319	7006	707943			473
10P200N8K5T	IRP	761486	93211	67702	0	922398	7	40	311
	IRP-T	775349	43205	34745	9111	862410			740
15P200N8K5T	IRP	873577	108408	54928	0	1036914	6	36	631
	IRP-T	881776	65903	21823	6399	975900			1003
20P200N8K5T	IRP	965250	103192	91284	0	1159726	3	41	645
	IRP-T	994712	87177	32218	10076	1124183			888
25P200N8K5T	IRP	1000991	102191	189719	0	1292900	12	40	520
	IRP-T	1002495	50638	80159	10077	1143370			338
30P200N8K5T	IRP	1102348	102335	100996	0	1305679	7	40	648
	IRP-T	1109641	73032	22481	11517	1216671			639

* Optimal solution

From Tables 2.6 and 2.7, we can see that, compared to IRP, IRP-T allows to reduce total costs with an average of 23% for small instances and 21% for large ones and enhances service levels with an average of 40% for small instances and 39% for large ones. Indeed, we can see that when transshipment is not considered, the supply chain experiences a high inventory and lost sales cost, as there is no emergency measure to sidestep shortage at the level of POS facing higher demands while others are overstocking (which increases idle stock and thus inventory cost). Thanks to transshipment and the M-M PD structure, excess stocks are dynamically reallocated while the system's cost is reduced. That is, the out-of-stock POS can reduce partially loss of sales (with an average of 66% for small instances and 63% for large ones) and thus enhance their service level, and the POS from which the transshipment is carried out can decrease inventory cost by reducing idle stocks that other POS can now use to meet demand at their levels (with an average of 31% for small instances and 27% for large ones). However, we can notice that, as expected, compared to IRP, transportation cost increases when transshipment is allowed as multiple pickups and deliveries are carried out (and more kilometres to be travelled). Moreover, for large instances in

same cases, although that inventory and lost sales costs are significantly reduced, the total cost is slightly reduced because the cost that transshipment generates (which can be seen as a penalty) cannot be offset by the savings it brings. Also, when the number of products is high, i.e. several demands of different sizes to be satisfied at each POS, it becomes more challenging to cover the costs generated by the transshipment: several pickups and deliveries that can take place at each POS. Therefore, transshipment can be of great interest once the costs it incurs can be offset by the savings it enables.

2.5.3.3 . Sensitivity analysis

Therefore, a sensitivity analysis is carried out because performances may depend on the different unit costs. Due to their significant contributions to the total costs, lost sale and transshipment unit costs are chosen to conduct the sensitivity analysis. Without loss of generality, we consider the instance 10P50N8K5T, a lost sale unit cost proportional to the holding unit cost: μh_{pi} ; where $\mu \in \{100; 150; 200; 250; 300\}$, and a transshipment unit cost: $b_{ij} = \nu \alpha d_{ij}$; where $\nu \in \{0.01; 0.03; 0.05; 0.07; 0.1\}$. Tables 2.8 and 2.9 provide the results regarding the breakdown of total costs. They also report cost-saving CS, improvement in service level SL and the corresponding \overline{CPU} time necessary to solve the IRP and IRP-T.

Table 2.8: Computational results for various values of lost sale unit cost

μ	Model	Cost (\$)					Improvement		$\overline{CPU}(s)$
		TR	INV	LS	TRA	TC	CS (%)	SL(%)	
100	IRP	29,810	32,833	38,627	0	101,270	11	22	535
	IRP-T	29,992	28,492	31,364	305	90,153			467
150	IRP	29,959	32,787	45,892	0	108,638	17	28	407
	IRP-T	30,066	26,616	32,960	512	90,154			490
200	IRP	29,806	32,895	53,010	0	115,711	46	32	485
	IRP-T	30,964	26,458	4,499	620	62,540			287
250	IRP	30,046	33,549	80,690	0	144,285	56	42	527
	IRP-T	32,922	23,317	6,210	1,025	63,474			352
300	IRP	30,035	35,546	103,410	0	168,990	60	53	494
	IRP-T	34,982	20,583	9,324	1,939	66,828			426

Table 2.9: Computational results for various values of LT unit cost

ν	Model	Cost (\$)					Improvement		$\overline{CPU}(s)$
		TR	INV	LS	TRA	TC	CS (%)	SL(%)	
0.01	IRP	29,806	32,895	53,010	0	115,711	46	32	485
	IRP-T	30,964	26,458	4,499	620	62,540			287
0.03	IRP	29,840	33,676	53,962	0	117,478	33	25	386
	IRP-T	30,520	28,044	19,539	563	78,665			507
0.05	IRP	29,940	32,860	53,227	0	116,028	10	17	454
	IRP-T	30,346	29,474	44,143	321	104,284			401
0.07	IRP	29,944	33,059	53,272	0	116,274	7	11	355
	IRP-T	30,072	31,722	46,761	123	108,677			575
0.1	IRP	29,975	33,038	53,948	0	116,961	1	4	298
	IRP-T	29,956	32,836	52,700	21	115,514			387

As expected, from Table 2.8, in the case of 10 products, when lost sale unit cost increases, it becomes binding to use transshipment as an additional

measure to reduce loss of sales and also to balance inventories between POS, which helps to reduce inventory holding cost. However, when the transshipment unit cost increases (see Table 2.9), fewer transshipment operations are performed as the cost it generates can no longer be offset by its savings. Therefore, we can conclude that the recourse to transshipment is due to stock shortages, and it might be a good practice to reduce the inventory costs. Moreover, trade-offs on unit costs are significant to safeguard cost reduction.

2.6 . Conclusions & Perspectives

This paper addresses a two-level supply chain in which a company manufactures products and sells them through its customers (own POS network). The problem of concern is a multi-product, multi-period, multi-vehicle inventory routing problem with transshipment under a deterministic demand and an ML policy. We formulate it as an MILP and exploit this formulation to find good initial solutions. We also design a two-phase matheuristic that integrates mathematical modelling strengthened with relevant derived valid inequalities and hybridises sophisticated metaheuristics: Genetic Algorithm and Simulated Annealing. The matheuristic was tuned with the parametrisation technique Irace. Experimentations were performed on new datasets specially designed for the IRP-T problem and are publicly available. Tested first on 660 best-known instances for a single and multi-vehicle IRP, the proposed algorithm proves to be efficient. Also, experiments highlight its limitations regarding large instances. Tested on the specially designed instances, the algorithm allows us to solve small-and-medium-sized instances in a reasonable amount of time and also solve the model for large instances up to 200 customers and 30 products. In what concerns the managerial insights, we demonstrate the benefits of promoting inventory sharing among different customers. This is a viable solution for supply chain managers aiming to improve the system's wide service level by dynamically reallocating (excess) stocks while lowering the system's cost. We show that it is in the company's interest to pool the customers' inventory and agree to manage all inventories collectively by entrusting the operations optimisation to the company's warehouse. The later optimises the total cost by deciding on the stock level of each customers and on the routes to be constructed over the planning horizon. LT occurs to fully avoid shortages. Lost sales are allowed when shortages occur and are used to measure the customer service level. The sensitivity analysis highlights the extent to which transshipment can be cost-effective by the savings it brings to reduce transportation costs, lost sales and inventory holding.

Finally, as a research outlook, it could be interesting to investigate the

benefits of LT under stochastic demands. Moreover, this work could be further developed by considering a realistic market, representing a competition between several POS, conflict of interest under joint control, pre-negotiated cash payments between either symmetric or asymmetric POS, and with different decision-makers. Nevertheless, depicting such markets requires a drastic mathematical transformation of the model. Specifically, it could differentiate between the decisions of the main stakeholders: CW, POS and carriers. Furthermore, as both direct emergency shipping and transshipment significantly contribute to environmental pollution, decisions related to fleet composition and fuel consumption alongside filling rate are to be considered in future work. Finally, as for the resolution approach, it might be worthwhile to check the performance of a pure metaheuristic algorithm to solve large problem instances and use other promising techniques to enhance its search performance.

3 - Modeling and solving a stochastic multi-product multi-vehicle IRP-T with product substitution

**Full Paper published in Applied Mathematical Modelling Journal, ISSN 0307904X, Impact Factor: 5.1, Quartile: Q1.
<https://doi.org/10.1016/j.apm.2021.08.029>**

Achamrah, F.E., Riane, F. and Limbourg, S. (2022). Spare parts inventory routing problem with transshipment and substitutions under stochastic demands. *Applied Mathematical Modelling*. 101, 309-331.

Spare parts inventory routing problem with transshipment and substitutions under stochastic demands

Fatima Ezzahra Achamrah ^{a,b}, Fouad Riane ^{a,b}, Sabine Limbourg ^c

^aComplex Systems and Interactions, Ecole Centrale of Casablanca; ^bLaboratoire Genie Industriel, CentraleSupélec, Paris Saclay University; ^c HEC-University of Liege (ULiege)

Abstract: We study a two-level spare parts supply chain in which a manufacturer supplies a central warehouse (CW) with original equipment manufacturer (OEM) and replacement or pattern parts (PP). The CW, distantly located from the manufacturer, distributes both OEM parts and PP to a given number of customers facing stochastic demands. The demand for spare parts is intermittent, exhibiting an infrequent rate and extreme dispersal overtime periods. Along with lateral transshipment, PP can be used as substitutes for the OEM parts to sidestep shortages at customers. Assuming that emergency shipments are significantly longer and more expensive, we aim to underline the relative effectiveness of such a new spare parts inventory management policy. A mixed-integer linear programming model is proposed to solve the inventory routing problem with transshipment and substitution under stochastic demands. The objective is to minimise costs of holding inventory and transportation, including regular shipment and transshipment, substitution and lost sales. To solve the problem, Sample Average Approximation method is used. Three demand patterns are studied based on empirical goodness-of-fit tests: Poisson distribution, stuttering Poisson distribution and negative binomial distribution. The model is tested on well-known benchmark instances generated for multi-product multi-vehicle IRP. Computational experiments highlight the benefits of promoting transshipment and substitution on the overall supply chain performance. Results also suggest insights that interest professionals willing to develop new decision support models for the most efficient management of such items.

Keywords: routing, transshipment, substitution, spare parts management, sample average approximation

3.1 . Introduction

Generally, in the first level of a spare parts supply chain, producers or manufacturers supply their customers with original equipment manufacturer (OEM) parts and/or aftermarket parts, also called replacement parts or pattern parts (PP). The latter are reverse-engineered OEM parts, designed to

perform in the same way as OEM parts, are less expensive, have a quality equivalent to or better than OEM parts, and are provided with a broader range of variety [Beiderbeck et al. \(2018\)](#); [Lacerda et al. \(2016\)](#). Spare parts customers such as distributors (or depots) and eventual re-sellers such as shops and car dealerships are on the second level of the supply chain. The critical role of the customers is to guarantee that end-user demands are met while incurring the lowest logistical costs [Ronzoni et al. \(2015\)](#).

An efficient spare parts inventory management is argued to be the backbone for reliable plant operations, cost reduction, and service level maximisation ([Upadhyay and Kumaraswamidhas, 2018](#)). Such items are at the most significant risk of obsolescence and may collectively account for up to 60% of the total stock value ([Johnston et al., 2003](#)). Moreover, their demand pattern is intermittent, exhibiting an infrequent rate and extreme dispersal over periods which often hinders the reduction of lost sales to customers. This is particularly true in the aerospace, IT, and automotive industry contexts ([Tur-rini and Meissner, 2019](#)). In this respect, the classical management models mainly designed to guarantee smooth replenishment often do not apply ([Syn-tetos et al., 2012](#)). Thus, alternative methods to manage inventory within the distribution network must be investigated to minimise logistics costs and ensure a high customer service level commitment [Van Jaarsveld and Dekker \(2011\)](#). This paper proposes a new management policy consisting of promoting multi-sourcing options to mitigate shortages: (1) regular shipment from manufacturer, (2) inventory sharing, or so-called lateral transshipment (LT) among customers, and (3) the use of PP as substitutes for OEM parts. We focus mainly on the inventory routing problem (IRP) that arises in a two-level spare parts supply chain in which a manufacturer distributes via her/his central warehouse (CW) a set of spare parts to a given number of customers facing stochastic demand. Indeed, assuming that emergency supplies from the CW are significantly long and expensive, this paper promotes spare parts substitution along with LT whenever demand exceeds the available stock at the level of each customer. That is, LT and substitution can be considered by customers to meet expected demands with the use of the same part from the inventory of other customers (LT) or with the use of a PP as substitutes for OEM parts held in their inventory (substitution).

Different incentives can promote the use of both LT and substitutions among customers. Regardless of the type of item, promoting LT is a viable solution for managers aiming to improve the system's broad service level by dynamically reallocating (excess) stocks while lowering the lost sales cost ([Paterson et al., 2011](#); [Coelho et al., 2012a](#); [Lefever et al., 2018](#)). Another criterion that can promote substitution is the ready availability of PP, which increases the likelihood of the user switching more often to. Not to mention the case where PP are of high quality. From the manufacturer's point

of view, it would be beneficial if she/he sells more PP than OEM parts by relying on branding and pricing, especially in a market with fewer substitute PP, which allows a higher probability of earning more significant profits. Moreover, multiple-sourcing is a promising research field for spare parts management (e.g., producing spare parts on-demand via additive manufacturing) (Minner, 2003; Sgarbossa et al., 2021). With this respect, the use of PP can be viewed as one of the more sourcing options available for spare parts management since PP is a less expensive and less reliable sourcing option. Another incentive for LT and substitution would be the slowness of the procedures regarding controlling the conformity of the imported spare parts to local or international standards. For instance, most automotive spare part distributors, such as Moroccan companies, often procure spare parts from domiciled offshore suppliers Achetoui et al. (2019a). The quality control procedure can take up to over 15 days and even longer if the Ministry's departments decide to rely on laboratories to analyse further the supplied parts Achetoui et al. (2019a,b). Moreover, spare parts are often stored until the results are published, which increases storage costs and renders emergency supplies significantly long and expensive Achetoui et al. (2019b).

On-demand distribution, parametric approaches rely upon a lead-time demand distributional assumption and an appropriate forecasting procedure for estimating the characteristics of a given distribution (i.e., means and variance) Syntetos et al. (2012); Turrini and Meissner (2019); Van der Auweraer and Boute (2019). For the case of fast-moving items, the Normality assumption is typically sufficient Lengu et al. (2014). However, Stock Keeping Units (SKUs) often exhibit intermittent or irregular demand patterns that may not be represented by the Normal distribution (Syntetos et al., 2012). This is almost invariably the case for spare parts since demands arise whenever a component fails or requires replacement instead of being generated according to buying behaviours of end-consumers Lengu et al. (2014); Syntetos et al. (2012). The reader is referred to (Boylan et al., 2008) for further details on such non-Normal demand patterns. Demand for spare parts exhibits thus an infrequent rate, extreme dispersal overtime periods, with some periods having no demand. In the literature, intermittent demand patterns are built from compound elements, namely a demand inter-arrival time and distribution of the demand sizes, when demand occurs (Syntetos et al., 2012; Turrini and Meissner, 2019; Van der Auweraer and Boute, 2019). As such, Syntetos et al. (2012) carry out an empirical analysis of the fitness of different compound distributions and their stock-control effects concerning inventories, demands and service levels in real-world contexts. According to the authors, the compound Poisson distribution (called the stuttering Poisson), a combination of a Poisson distribution for demand occurrence and a geometric distribution for demand size, outperforms in all considered data

sets. In this paper, three compound distributions are considered. In addition to stuttering Poisson distribution, we conduct experiments for two other distributions: Poisson distribution for demand occurrence, combined with demands of constant size and negative binomial distribution.

This paper contributes to the literature in three main dimensions. It studies a two-level spare parts supply chain where a manufacturer supplies a CW with OEM and PP parts. The CW, distantly located from the manufacturer, distributes both OEM parts and PP to a given number of customers facing stochastic demands. These customers may thus form a virtual pool of their parts inventories, allowing LT. Unlike other research, our approach integrates LT decisions in designing routes that carry out regular shipments from CW. Substitutions are also possible among parts for which waiting is not an option. In addition, Parts are assumed to be substitutable only if they have the same shape, fit, and function. Substitution is also considered to be not bi-directional. That is, part 1 is, for example, substituting part 2, and the inverse is not necessarily applied. Considering the aforementioned multi-sourcing options and assuming that emergency shipments are significantly longer and more expensive, we aim to underline the relative effectiveness of the spare parts inventory management policy based on LT and substitutions. We model the problem as a two-stage stochastic multi-product multi-vehicle IRP considering LT and substitution as emergency measures to mitigate shortages. The objective is to minimise the total cost, including the inventory holding cost at the CW and customers, transportation which includes regular shipment and transshipment, substitution and lost demands. The sample average approximation method is used because of its good convergence properties to solve the problem. Based on empirical goodness-of-fit tests of (Syntetos et al., 2012), three different demand patterns are studied: Poisson distribution for demand occurrence, combined with demands of constant size, stuttering Poisson distribution and negative binomial distribution. The model is tested on well-known benchmark instances generated for multi-product multi-vehicle IRP. Computational experiments provide insights into the benefits of promoting transshipment and substitution on the overall supply chain performance. They also suggest findings that may interest practitioners willing to improve decision support models for the most effective management of such items.

The remainder of the paper is structured as follows. Section 2 presents related works. After a detailed description of the problem in Section 3, a mathematical formulation and a solution approach are provided in Section 4. Section 5 reports computational results. Finally, section 6 presents conclusions and perspectives.

3.2 . Related work

The classical IRP includes inventory management, vehicle routing, and delivery scheduling decision problems (Coelho and Laporte, 2014). Such decisions can be streamlined by introducing a vendor-managed inventory (VMI) approach, which incorporates replenishment and distribution processes, resulting in overall logistics cost reduction. The deterministic versions of the IRP have been widely studied. Applications can be found in (Archetti et al., 2007; Geiger and Sevaux, 2011; Coelho and Laporte, 2013a; Christiansen et al., 2011; Zhao et al., 2008). The most closely aligned works to this paper are the ones that studied the stochastic variant of the IRP. The Stochastic IRP (SIRP) is similar to the deterministic IRP, except that the customer's demand is known in a probabilistic sense (Coelho and Laporte, 2014). More recent works on SIRP include the one of (Bertazzi et al., 2013), in which stockouts and finite horizon SIRP are studied and solved using a dynamic programming model and a hybrid roll-out algorithm. A similar problem is addressed in a robust optimisation approach through MILP formulations by (Soysal et al., 2015), who suggest a robust-based strategy for these demands that assumes a uniform random behaviour. In Huang and Lin (2010), the authors develop a modified ant colony optimisation metaheuristic for the multi-product SIRP. Considering stochastic customer demands and replenishment lead-times, a robust inventory routing policy is addressed in (Li et al., 2016). Dynamic SIRP under Maximum Level (ML) and Order Up-To level (OU) policies are studied in (Coelho et al., 2014a). Authors use a proactive and reactive approach to solve the problem at hand. In (Roldán et al., 2016), this work is extended by addressing the robustness of inventory replenishment and customer selection policies. In Yu et al. (2012), a SIRP with split deliveries and service level constraints is addressed. In Rahim et al. (2014), a multi-period IRP with stochastic stationary demand through a deterministic equivalent approximation model is studied. Finally, in (Gruler et al., 2018) the authors study a SIRP and incorporate constructive components in a simheuristic they use to solve the problem.

It is in the latter context that Coelho et al. (2012b) introduce the concept of LT between customers within a deterministic inventory routing (IRP-T). Coelho et al. (2012b) study a single-product, single-vehicle IRP-T and use an Adaptive Large Neighbourhood Search heuristic (ALNS) to solve large-scale instances. The authors assume that shortages are not permitted, and LT is performed by a carrier's vehicles and not by the supplier. In Lefever et al. (2018), the authors extend the work of Coelho et al. (2012b) by proposing new sets of valid inequalities to strengthen the linear relaxation. A multi-product IRP-T is studied and solved using a Randomised Variable Neighbourhood Descent in Peres et al. (2017). On stochastic IRP-T (SIRP-T), (Coelho et al., 2014a) study the stochastic version of the problem addressed in Coelho et al. (2012b). Under the same assumptions, the authors propose a reac-

tive and proactive policy to solve the single-product, single-vehicle SIRP-T. In [Tarhini et al. \(2019\)](#), a multi-product, multi-vehicle SIRP-T is addressed and solved using a Variable Neighbourhood Search algorithm. In [Dehghani et al. \(2021\)](#), a mathematical model that decides on proactive transshipment under stochastic demand to reduce total costs and shortages in a blood supply chain is developed. Finally, single-product SIRP-T is studied in [Fattahi and Tanhatalab \(2021\)](#). Lot sizing and perishability of the product are also considered. The authors propose a Lagrangian relaxation-based heuristic to solve the problem at hand.

Based on this literature review, it can be stated that fewer papers on IRP-T and SIRP-T study a multi-product multi-vehicle version and take into account lost sales due to the shortages as a measure of the service quality. Moreover, in the design of vehicle routing, LT-related decisions are not integrated as LT is always assumed to be subcontracted or handled by another carrier's vehicles. Indeed, to simplify the optimisation problem, authors identify just the nodes and periods when LT may take place and manage stocks so LT may be performed. In this respect, this paper is thus intended to fill this gap. Furthermore, to the best of the authors' knowledge, none of the existing papers incorporates product substitution within the settings and promotes LT between customers to avoid shortages of parts. We study the following a multi-product multi-vehicle SIRP-T and ML inventory policy. We develop an appropriate model to underline the relative effectiveness of this new spare parts inventory management policy based on LT and substitutions and use average sample approximation (SAA) to solve it.

3.3 . Problem setting

We consider a spare parts supply chain with two levels. CW distributes different parts to a certain number of customers (the second level). Compared to the distance from the CW, the customers are located at a negligible distance away from each other, and all hold low-demand spare parts. By allowing LT, these customers form a pool that can share inventories. In this paper, we assume that LT is not outsourced and incorporated into the routes carrying regular shipments. We also assume that PP can also be transshipped between customers. Given the demand distributions of customers, at the beginning of the planning horizon, the CW's manager needs to choose the routes, inventory levels, transshipment, and substitution decisions. Once demand is realised if it exceeds the available capacity at the level of each customer, spare part substitution and LT are used as a recourse. That is, when critical OEM parts are unavailable, transshipped parts and the compatible PP can be used to satisfy the customers' demand. As is commonly the case, we assume that the vehicle capacity is expressed as a function of the

demands to be satisfied (Archetti et al., 2007; Coelho and Laporte, 2013a, 2014). We also assume that all PP made available to customers are reliable and of good quality. Parts are substitutable only if they have the same shape, fit, and function. We assume that the substitution is not bi-directional. We also implicitly assume that emergency supplies from the CW are significantly long and expensive. Therefore, LT and substitutions are favoured over the use of emergency supplies. Customers can then consider transshipment and substitution to meet expected demands using the same part from the inventory of other locations (transshipment) or using a compatible part from their inventory (substitution).

The problem of concern can be addressed as a two-stage stochastic inventory-routing problem considering transshipment and substitution as emergency measures to mitigate shortages. We assume the routing (which customers to visit in each period) is the first-stage decision. The quantities to deliver, including transshipment, the lost sales at each period, and the inventory levels and substituted quantities are adjusted to the scenario. The goal of the stochastic approach is to find the solution that minimises the routing cost plus the expected cost of both inventory, transshipment, substitution and shortage due to loss of sales.

3.4 . Mathematical formulation

3.4.1 . Mathematical modelling

The SIRP-TS is defined on a graph $G = (\mathcal{N}, \mathcal{A})$, where \mathcal{N} is the vertex set indexed by $i \in \{0, \dots, n\}$ and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$ is the edge set. Vertex 0 represents the CW, and the set $\mathcal{N}_0 = \mathcal{N} \setminus \{0\}$ denotes the customers. The length of the planning horizon is \mathcal{H} with discrete time periods t . Each customer $i \in \mathcal{N}_0$ has demands for spare part $p \in \{1, \dots, m\}$ per period $t \in \mathcal{H} = \{1, \dots, T\}$ which is a random variable D_{pit} . A scenario, denoted ω , is a set of potential demands that appear by the end of the horizon \mathcal{H} . We denote the set of scenarios for the realisation of demands by Ω . Thus, $D_{pit}(\omega)$ denotes the demand of a customer i of a spare part p in period t in scenario $\omega \in \{1, \dots, |\Omega|\}$. Moreover, each customer and the CW, $i \in \mathcal{N}$, incur unit inventory holding costs, h_{pi} , per period and per spare part $p \in \{1, \dots, m\}$, with inventory capacities K_i . Inventories are not allowed to exceed the holding capacity and must be positive. At the beginning of the planning horizon, at each location $i \in \mathcal{N}$, the current inventory levels I_{pi0} of the spare part p are known. A set of homogeneous vehicles $v \in V = \{1, \dots, k\}$ is available, each with a capacity Q in terms of the spare part without distinction between them, with a being a fixed transportation cost per km. Each vehicle is able to perform a route per period. A distance d_{ij} is associated for all $(i, j) \in \mathcal{A}$. A transshipment can

start from any customer, i.e., a customer can transfer to other locations as needed. transshipment can occur when it is profitable to ship spares between customers. As in [Coelho et al. \(2012b\)](#), we choose a transshipment cost per unit of $b_{ij} = 0.01 a d_{ij}$. The unit cost of substituting a spare part p by $s \in P$ is c_{ps} . All possible combinations according to the parts' compatibility are represented by o_{ps} , which is equal to 1 if a spare part p is compatible with a spare part s , and 0 otherwise. The lost sales cost associated with a spare part p at the customer i is f_{pi} , and the quantity of spare parts shipped by the manufacturer (from the factory) to the CW during period t is g_{pt} . Finally, it is assumed that holding and vehicle capacities, and quantity shipped by the manufacturer are exogenous parameters and are not under the control of the CW. All of these notations as well as decision variables are summarised in Table 3.1:

Table 3.1: Notation summary

Sets	
\mathcal{N}	Set of nodes including CW
\mathcal{N}_0	Set of customers
\mathcal{H}	Planning horizon indexed by t
V	Set of vehicles indexed by v
\mathcal{P}	Set of spare parts indexed by p
Ω	Set of scenarios indexed by ω ; $\omega \in \{1, \dots, \Omega \}$
Parameters	
a	Transportation cost per km
d_{ij}	Distance in km between $(i, j) \in \mathcal{A}$
h_{pi}	Unit inventory holding cost per period for spare part p at node $i \in \mathcal{N}$
b_{ij}	Unit transshipment cost between node $i \in \mathcal{N}$ and node $j \in \mathcal{N}$
K_i	Maximum inventory capacity at node $i \in \mathcal{N}$
I_{pi0}	Inventory level of a spare part p at location $i \in \mathcal{N}$ at the beginning of the planning horizon
Q	Vehicle capacity
c_{sp}	Unit cost associated with the substitution of a spare part p by a substitute s
o_{sp}	Equal to 1 if a substitute part s is compatible with a spare part p , and 0 otherwise
Variables	
x_{ijvt}	Equal to 1 if $(i, j) \in \mathcal{A}$ is visited by the vehicle v in period t , 0 otherwise
u_{vt}	Equal to 1 if the vehicle v is used in period t , 0 otherwise
$I_{pit}(\omega)$	Inventory level of a spare part p at a location $i \in \mathcal{N}$ at the end of period t for a scenario ω
$Q_{pit}(\omega)$	Quantity of a spare part p delivered from the CW to the customer $i \in \mathcal{N}_0$ in a period t for a scenario ω
$q_{pijvt}(\omega)$	Quantity of a spare part p transported from a location $i \in \mathcal{N}$ to a location $j \in \mathcal{N}$ by vehicle v in a period t for a scenario ω . This quantity includes regular shipment from CW and transshipment between customers
$y_{pijvt}(\omega)$	Quantity of a spare part p transshipped from the customer $i \in \mathcal{N}_0$ to the customer $j \in \mathcal{N}_0$ by vehicle v to address a shortage of spare p in a period t for a scenario ω
$w_{pit}(\omega)$	Lost sales quantity of a spare part p at the customer $i \in \mathcal{N}_0$ in a period t for a scenario ω
$z_{spit}(\omega)$	Quantity of a spare part s substitute for spare part p used at the customer i in period t for a scenario ω to satisfy a part of the unsatisfied demand for $D_{pit}(\omega)$ and for all $o_{sp}=1$

The formulation of the SIRP-TS can be written as:

$$\begin{aligned}
\min \quad & \sum_{t \in \mathcal{H}} \sum_{v \in V} \sum_{i, j \in \mathcal{N}} a d_{ij} x_{ijvt} + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left[\sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}} \sum_{p \in \mathcal{P}} h_{pi} I_{pit}(\omega) + \right. \\
& \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{H}} \sum_{v \in V} \sum_{i, j \in \mathcal{N}_0} b_{ij} y_{pijvt}(\omega) \\
& \left. + \sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}_0} \sum_{p \in \mathcal{P}} f_{pi} w_{pit}(\omega) + \sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}_0} \sum_{p, s \in \mathcal{P}} c_{sp} z_{spit}(\omega) \right] \quad (3.1)
\end{aligned}$$

Subject to:

$$I_{pit}(\omega) = I_{pit-1}(\omega) + Q_{pit}(\omega) - D_{pit}(\omega) + w_{pit}(\omega) +$$

$$\begin{aligned} & \sum_{v \in V} \sum_{j \neq i \in \mathcal{N}_0} (y_{pjivt}(\omega) - y_{pijvt}(\omega)) \\ & + \sum_{s \neq p \in \mathcal{P}} (z_{spit}(\omega) - z_{psit}(\omega)) \quad \forall p \in \mathcal{P}, i \in \mathcal{N}_0, t \in \mathcal{H} \end{aligned} \quad (3.2)$$

$$I_{p0t}(\omega) = I_{p0t-1}(\omega) - \sum_{i \in \mathcal{N}} Q_{pit}(\omega) + g_{pt} \quad \forall p \in \mathcal{P}, t \in \mathcal{H} \quad (3.3)$$

$$\begin{aligned} Q_{pj}(\omega) &= \sum_{v \in V} \sum_{i \neq j \in \mathcal{N}_0} (y_{pijvt}(\omega) - y_{pjivt}(\omega) + q_{pijvt}(\omega) - q_{pjivt}(\omega)) \\ & \quad \forall p \in \mathcal{P}, j \in \mathcal{N}_0, t \in \mathcal{H} \end{aligned} \quad (3.4)$$

$$\sum_{p \in \mathcal{P}} q_{pi0vt}(\omega) = 0 \quad \forall i \in \mathcal{N}_0, v \in V, t \in \mathcal{H} \quad (3.5)$$

$$\sum_{p \in \mathcal{P}} I_{pit}(\omega) \leq K_i \quad \forall i \in \mathcal{N}, t \in \mathcal{H} \quad (3.6)$$

$$\sum_{p \in \mathcal{P}} q_{pijvt}(\omega) \leq Qx_{ijvt} \quad \forall i, j \in \mathcal{N}, v \in V, t \in \mathcal{H} \quad (3.7)$$

$$\sum_{p \in \mathcal{P}} q_{pijvt}(\omega) \leq Qu_{vt} \quad \forall i, j \in \mathcal{N}, v \in V, t \in \mathcal{H} \quad (3.8)$$

$$\sum_{v \in V} \sum_{j \neq i \in \mathcal{N}_0} y_{pijvt}(\omega) \leq I_{pit-1}(\omega) \quad \forall p \in \mathcal{P}, i \in \mathcal{N}_0, t \in \mathcal{H} \quad (3.9)$$

$$\sum_{i \neq j \in \mathcal{N}} x_{ijvt} = \sum_{i \neq j \in \mathcal{N}} x_{jivt} \quad \forall j \in \mathcal{N}, v \in V, t \in \mathcal{H} \quad (3.10)$$

$$\sum_{i \neq j \in \mathcal{N}} \sum_{v \in V} x_{ijvt} \leq 1 \quad \forall j \in \mathcal{N}_0, t \in \mathcal{H} \quad (3.11)$$

$$\sum_{j \in \mathcal{N}_0} x_{0jvt} = u_{vt} \quad \forall v \in V, t \in \mathcal{H} \quad (3.12)$$

$$\sum_{v \in V} u_{vt} \leq k \quad \forall t \in \mathcal{H} \quad (3.13)$$

$$y_{pijvt}(\omega) \leq x_{ijvt} K_j \quad \forall i, j \in \mathcal{N}, p \in \mathcal{P}, v \in V, t \in \mathcal{H} \quad (3.14)$$

The objective function (3.1) minimises the total cost. The first term corresponds to the transportation costs, the second term to the inventory cost at the CW and customers' locations, the third term to the transshipment costs, the fourth term to lost sales costs at the customers' locations, and the last term corresponds to the cost of substitutions. Constraint (3.2) indicates that for each customer i and for each spare part p , the inventory level at period t is the inventory level at the previous period plus the delivered quantity of the spare part p and the lost sales w_{pit} if any, minus the demand, plus the difference between the quantity of spare part p transshipped to and from i , plus the difference between the quantity of spare part s used as a

substitute of p and the quantity of p used as a substitute of the other spare parts. Constraint (3.3) expresses the conservation conditions of inventory at the CW over successive periods. The conditions take into account quantities delivered to the CW and to the customers. Constraint (3.4) expresses the flow conservation conditions at a customer j . Constraint (3.5) states that at the end of each period, vehicles must return back empty to the CW. Constraint (3.6) guarantees that inventory levels do not exceed the maximal available inventory capacity. Constraints (3.7) and (3.8) state that the maximum capacity of the vehicle is not exceeded. Constraint (3.9) states that the quantity transshipped from a customer i at a period t does not exceed the initial inventory level at this period. Constraint (3.10) stipulates that if a vehicle v visits the customer j , it must leave j in the same period t . Constraint (3.11) ensures that, at most, one vehicle v visits a customer per period. Constraint (3.12) ensures that only vehicles carrying parts leave the CW. Constraint (3.13) stipulates that the sum of vehicles used in time period t is bounded by the number of available vehicles. Constraints (3.14) ensure that spare part p is transshipped from the node i to the node j by vehicle v if the arc (i, j) is being used by the vehicle v during the period t .

3.4.2 . Sample Average Approximation

Given their inherent analytical complexities and high computational requirements, solving large-scale stochastic optimisation problems is highly challenging (Kleywegt et al., 2002; Shastri and Diwekar, 2006). The SAA method's good convergence properties, which have been thoroughly explored in the literature, are one of its most appealing features (Banholzer et al., 2019; Jiang et al., 2021). With regards to SAA estimators' consistency, which is often seen as a minimum criterion on any excellent estimator, (Dupacová and Wets, 1988) stress, in a somewhat general fashion, that the sequence of approximate objective function epi-converges to the optimal solution. This enables the inference of sets of optimal values with high consistency. An alternate approach has been followed based on the epi-convergence. This approach draws from a strong consistency of optimal estimators by constructing almost sure uniform convergence (Shapiro et al., 2014). In (Banholzer et al., 2019), authors examine the rates at which optimum SAA estimators converge in the nearly certain and mean sense with their deterministic counterparts. Finally, in Verweij et al. (2003), the SAA method is applied to three classes of 2-stage stochastic routing problems. Through considerable experimentation, the authors proved the excellent convergence properties of SAA and the high quality of solutions to the stochastic programming problems under consideration. The reader, if interested, is referred to this paper for further details. For all these reasons, SAA has been widely used to cover a large variety of applications such as stochastic supply chain design and optimisation problems of large scale (Santoso et al., 2005); stochastic knapsack

problem (Kleywegt et al., 2002) and reliability-based design of engineering systems (Royset and Polak, 2004).

In SAA, the objective function value of the stochastic problem is unknown and approximated using a random sample estimate (Homem-de Mello and Bayraksan, 2014; Royset and Szechtman, 2013). The objective function is evaluated iteratively for a given number of scenarios before the optimality gap falls below a particular threshold value. SAA provides a straightforward framework conducive to parallel implementation and reduction of variance techniques. It also possesses good convergence properties and well-developed statistical methods to validate solutions and perform error analysis. For this reason, SAA is used to solve the SIRP with transshipment and Substitution (SIRP-TS).

SAA approximates the expected cost of the objective function by the average sample function. This expected cost is replaced by the mean value of a random set of samples $(\omega^1, \omega^2, \dots, \omega^{|\Omega|})$ of size $|\Omega|$ obtained by the Monte Carlo method, where Ω is the set of ω -indexed scenarios. Each time, this is repeated L times with different samples, resulting in a candidate solution. Thus, SAA method generates L separate sample sets $\Omega_l, l \in \{1, \dots, L\}$. For each scenario set, Ω_l , the related SAA problem (where Ω is replaced by Ω_l in SIRP-TS) is solved and generates a candidate solution. Therefore, the first-stage solution is fixed for each candidate solution, and the value of the objective function for a very large sample with l scenarios is computed. This value is computed in the case of the two-stage model SIRP-TS by solving a pure linear programming problem on the second-stage variables.

With a reasonable level of accuracy, SAA solves the real problem if certain conditions are met (Kleywegt et al., 2002; Shapiro and Philpott, 2007). These requirements and justifications for how SIRP-TS meets them are as follows:

1. a sample realisation of the random variable can be generated. For SIRP-TS, this can be done for each random variable D_{pit} which represents demand each customer i has to satisfy for each spare part p and for each period t (see Section 3.5.1.2).
2. with moderate sample size, the SAA problem can be solved effectively. In the computational experiments section, we will show that for most test instances, with a sample size of 20, we can solve SIRP-TS in a reasonable amount of time.
3. the expected costs can be easily calculated by solving the model for a given first-stage solution and a given realisation of demand.
4. there is a complete recourse to the actual problem, i.e. every solution to the first stage problem is feasible for the second stage. In SIRP-TS,

this is made possible by assuming that transshipment and substitutions are always used when demand cannot be met with the related first-stage variables.

Statistical estimates of the lower and upper bounds on the objective function value of the stochastic problem, as well as estimates of the variances of these bounds, can be computed to evaluate the quality of the SAA solution (Kleywegt et al., 2002; Shapiro and Philpott, 2007). Let \hat{X}^l be a candidate solution to the first stage with an objective function \hat{F}^l . To estimate the lower bound of the true objective function value, the mean \bar{F} and the variance $\hat{\sigma}_{\Omega,L}^2$ of the objective function $\hat{F}^1, \dots, \hat{F}^L$

$$\bar{F} = \frac{1}{L} \sum_{l=1}^L \hat{F}^l \quad (3.15)$$

$$\hat{\sigma}_{\Omega,L}^2 = \frac{1}{L(L-1)} \sum_{l=1}^L (\hat{F}^l - \bar{F})^2 \quad (3.16)$$

The lower bound is then expressed as:

$$LB = \bar{F} - \chi_{\pi,\rho} \hat{\sigma}_{\Omega,L} \quad (3.17)$$

where $\chi_{\pi,\rho}$ is the π -critical value of the χ -distribution with ρ degrees of freedom, $\rho = L - 1$

By assessing the solution with a very large scenario tree of size $|\Omega'|$ that is assumed to represent the true distribution of demand, the upper bound on the true objective function value of each candidate solution is computed. As each scenario $\omega \in \{1, \dots, |\Omega'|\}$ is an i.i.d. random sample, the problem of assessing a candidate solution is broken down into Ω' sub-problems. The size of Ω' which is the scenario tree is far larger than the one held in any SAA run (Kleywegt et al., 2002). The objective function value of a given sub-problem ω is denoted as $\psi_{\omega}(X^l)$, which is calculated as:

$$\begin{aligned} \psi_{\omega}(X^l) = & \sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}} \sum_{p \in \mathcal{P}} h_{pi} I_{pit}(\omega) + \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{H}} \sum_{v \in V} \sum_{i,j \in \mathcal{N}_0} b_{ij} y_{pijvt}(\omega) \\ & + \sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}_0} \sum_{p \in \mathcal{P}} f_{pi} w_{pit}(\omega) + \sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}_0} \sum_{p,s \in \mathcal{P}} c_{sp} z_{spit}(\omega) \end{aligned} \quad (3.18)$$

It has be noted that, $|\Omega'|$ can be very large since each subproblem is solved separately without creating a significant computation complexity. The approximation of the true objective value, denoted as $\psi(X^l)$, of the second stage problem is computed as:

$$\psi(X^l) = \frac{1}{|\Omega'|} \sum_{\omega=1}^{|\Omega'|} \psi_{\omega}(X^l) \quad (3.19)$$

The value of the true objective function, \bar{F}^l , for a candidate \hat{X}^l and its variance $\hat{\sigma}_{|\Omega^l|}^2(X^l)$ as computed then as follows:

$$\bar{F}^l = \psi(X^l) + \sum_{t \in \mathcal{H}} \sum_{v \in V} \sum_{i,j \in \mathcal{N}} ad_{ij} x_{ijvt} \quad (3.20)$$

$$\hat{\sigma}_{|\Omega^l|}^2(X^l) = \frac{1}{|\Omega^l|(|\Omega^l|-1)} \sum_{\omega=1}^{|\Omega^l|} (\psi_{\omega}(X^l) - \psi(X^l))^2 \quad (3.21)$$

The upper bound of the candidate \hat{X}^l , ϕ^l is then computed as:

$$\phi^l = \bar{F}^l - \tau_{\pi} \hat{\sigma}_{|\Omega^l|}^2(X^l) \quad (3.22)$$

where τ_{π} is the π -critical value of the τ -distribution. The upper bound of the algorithm is the smallest ϕ^l and the candidate solution \hat{X} refers to the solution with the smallest upper bound solution that results in the smallest optimality gap for all candidate solutions:

$$\hat{X} = \underset{l \in \{1, \dots, |L|\}}{\operatorname{argmin}} (\phi^l) \quad (3.23)$$

3.5 . Computational experiments

This section presents the experimental design adopted in this paper as well as the computational results.

3.5.1 . Experimental design

3.5.1.1 . SAA setting

To solve the SIRP, $|\Omega|=20$ scenarios are used to compute the expected costs of the second stage. Each scenario is repeated $L=20$ times to calculate the LB on the expected value. This choice is based on a trade-off between the optimality gap and computational time. For the UB on expected cost, all L candidate solutions are assessed using a scenario tree of $|\Omega^l|=800$. Finally, all optimisation steps are carried out with a personal computer (MacBook Pro, macOS Big Sur, 3.3 GHz Quad-Core Intel Core i7 CPU with 8 GB of RAM) and with CPLEX 12.9 and Python 3.7. A maximum time limit of 1200 seconds is fixed.

3.5.1.2 . Demand distributions

The demand for spare parts occurs when a component fails or needs replacement rather than triggered by the end-user purchasing behaviours (and the way demand moves upstream in a supply chain). It is possible, then, to identify such items as sporadic and slow movers arising at irregular intervals and variable sizes. It is preferable to model spare parts demand from the constituent components, i.e. the demand size and the inter-demand interval.

Consequently, compound theoretical distributions (which specifically include the combination of size and interval) are widely used in such application contexts (Conceição et al., 2015; Syntetos et al., 2012; Turrini and Meissner, 2019). In this paper, demands each customer i has satisfy per spare part p and per period t are random variables D_{pit} . Different distributions have been studied based on empirical goodness-of-fit tests. Discrete distributions are chosen since they provide a better fit for intermittent demands compared to continuous ones. According to Syntetos et al. (2012), these distributions are (1) Poisson distribution (PD) for demand occurrence with demands of constant size; (2) stuttering Poisson distribution (SPD), with Poisson distribution for demand occurrence and a geometric distribution for demand size; and (3) negative binomial distribution (NBD), with a Poisson distribution for demand occurrence and logarithmic distribution for demand size.

For $\beta = 0,1,2, \dots$, the distribution functions of Poisson distribution occurrence $PD_\lambda(\beta)$ can be expressed as:

$$PD_\lambda(\beta) = \frac{\lambda^\beta e^{-\lambda}}{\beta!} \quad (3.24)$$

the stuttering Poisson distribution $SPD_{(\lambda,\theta)}(\beta)$ as

$$SPD_{(\lambda,\theta)}(\beta) = \sum_{1 \leq i \leq \beta} e^{-\lambda} \frac{\lambda^i}{i!} \binom{\beta-1}{i-1} \theta^i (1-\theta)^{\beta-i} \quad (3.25)$$

where λ and θ are the Poisson and geometric distribution parameters, and the Negative Binomial distribution $NBD(r, \mu)(\beta)$ as:

$$NBD(r, \mu)(\beta) = \binom{\beta+r-1}{\beta} \mu^r (1-\mu)^\beta \quad (3.26)$$

where r is the number of successes, and μ is the probability of success.

To generate an independent and identical distributed (i.i.d.) random sample of $|\Omega|$ realisations of D_{pit} for each distribution under consideration, the Inverse Transform Sampling algorithm 3 is used.

Algorithm 2 Inverse Transform Sampling

- 1: **procedure** ITS(F) ▷ F is a distribution function
 - 2: $\alpha \leftarrow$ Generate random number
 from the standard uniform distribution in $[0, 1]$;
 - 3: $\beta \leftarrow F^{-1}(\alpha)$
 - 4: **end procedure**
-

3.5.1.3 . Other input data

First, the model is tested on randomly instances generated by [Coelho and Laporte \(2013a\)](#) for multi-product multi-vehicle IRP. Following a brief description is provided, and the reader is referred to their paper for further details. The dataset can be downloaded from <http://www.leandro-coelho.com/instances/>. For each instance, the number of customers varies between 10 and 50, and the number of products and vehicles varies between 1 and 5. Each instance contains 3, 5 and 7 periods. Product availability at the CW is a multiple of a number randomly generated according to a discrete uniform distribution in the interval $[50, 140]$, and the maximum inventory level is a multiple of a number drawn randomly from $[150, 200]$. The initial inventory level is a randomly generated number in the interval $[100, 150]$. Holding costs are randomly generated from a continuous uniform distribution in the interval $[0.02, 0.2]$. As in [Coelho et al. \(2014a\)](#) shortage penalty cost equals 200 times the holding cost.

Secondly, to highlight the benefit of promoting substitutions and transshipment on the overall supply chain performance, we use the same set of instances but this time for a number of products varying between 20 and 40. For each customer, period and spare part Poisson and geometric distribution parameters λ and θ , as well as the NBD parameters r and μ , are random numbers generated between 0 and 1.

An instance name is referred to as n [number of customers] m [number of spare parts] k [number of vehicles] T [number of periods], e.g. n5m20k2T5 is an instance consisting of 5 customers, demands for a number of spare parts equal to 20 performed by two vehicles in a planning horizon that corresponds to 5 days.

3.5.2 . Computational experiments

This section presents each computational distribution result obtained for all instances under consideration. Detailed results are shown in Appendix B. For an illustrative purpose, Table 3.2 reports costs computed for products varying between 1 and 5 and the number of customers equal to 10. It provides the first and second stage (FSC, SSC), standard deviation regarding the upper and lower bound (UB, LB), and CPU time in the second. Table 3.3 reports for all instances under consideration the breakdown of costs, namely: Transportation (T), Inventory (I), Lost sales (LS), substitution (S) and transshipment (Ts). All experiments are performed for four different models: SIRP, SIRP with Transshipment (SIRP-T), SIRP with Substitution (SIRPS) and finally, SIRP-TS. They also report cost-saving (CS) computed concerning total cost (TC), which is expressed as follows:

$$CS = \frac{TC_{SIRP} - TC_{SIRP-with-X}}{TC_{SIRP}} \quad (3.27)$$

Table 3.2: Computational results for a number of product varying between 1 and 5 - number of customers equal to 10

Instances	Model	PD					SPD					NBD				
		FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)
n10m1k3T3	SIRP	1590.2	2512.6	0.01	0.09	30	2116.6	2921.7	0.06	0.07	36	2409.4	3868.3	0.04	0.09	45
	SIRP-T	1436.8	2091.6	0.07	0.00	45	1696.1	2456.9	0.03	0.08	50	2160.0	3031.1	0.02	0.05	63
n10m1k3T5	SIRP	3430.0	5100.2	0.08	0.07	82	3719.8	5825.5	0.10	0.01	91	3869.7	7956.9	0.09	0.05	114
	SIRP-T	2995.1	4511.5	0.07	0.01	75	3105.2	5339.7	0.01	0.05	90	3480.2	6273.7	0.03	0.04	121
n10m1k3T7	SIRP	4820.3	7291.1	0.07	0.01	116	5741.6	8961.6	0.01	0.04	129	7043.9	9908.9	0.05	0.02	161
	SIRP-T	4255.8	6402.3	0.08	0.03	113	4483.3	7389.8	0.06	0.05	124	5655.1	8117.7	0.01	0.01	160
n10m3k3T3	SIRP	1928.0	2722.3	0.09	0.01	48	2219.4	3430.7	0.01	0.08	63	2506.8	3764.9	0.10	0.06	78
	SIRP-T	1572.1	2427.1	0.03	0.04	34	1914.3	2896.8	0.04	0.09	37	2385.0	3378.7	0.08	0.10	46
	SIRPS	1618.0	1567.2	0.08	0.06	60	2143.6	1945.7	0.09	0.06	74	1955.8	3774.7	0.04	0.07	92
	SIRP-TS	1686.2	1141.5	0.09	0.01	75	2158.9	905.3	0.02	0.10	97	2604.9	1710.1	0.01	0.02	120
n10m3k3T5	SIRP	3768.4	5835.1	0.06	0.09	89	4589.7	7145.7	0.09	0.06	109	5677.9	9073.6	0.00	0.06	134
	SIRP-T	3401.3	4953.7	0.07	0.07	68	4112.0	6031.0	0.03	0.09	84	4500.6	7549.3	0.01	0.08	102
	SIRPS	3537.7	5009.5	0.04	0.06	41	3508.1	6201.5	0.05	0.00	47	4657.0	7489.6	0.00	0.03	59
	SIRP-TS	3463.2	4891.9	0.06	0.06	58	3677.1	4513.4	0.03	0.04	71	4276.2	5934.5	0.05	0.02	92
n10m3k3T7	SIRP	5344.0	7946.3	0.06	0.09	87	6189.7	9971.3	0.09	0.03	104	7173.8	11282.1	0.02	0.05	139
	SIRP-T	4649.3	7311.9	0.06	0.01	111	5194.3	8489.4	0.08	0.01	133	5751.7	9916.2	0.05	0.09	178
	SIRPS	4655.1	6907.5	0.07	0.04	75	5397.3	7518.0	0.02	0.02	89	5725.0	9553.9	0.06	0.01	109
	SIRP-TS	4519.3	6910.4	0.08	0.02	74	5263.8	5669.0	0.05	0.07	85	5486.2	8536.5	0.04	0.08	105
n10m5k3T3	SIRP	2427.0	3663.3	0.06	0.06	69	2911.1	4068.3	0.06	0.01	79	3107.2	5009.9	0.06	0.02	101
	SIRP-T	2212.8	3268.5	0.02	0.01	63	2654.7	3939.2	0.04	0.03	71	2873.5	4870.3	0.02	0.03	89
	SIRPS	2010.2	3227.4	0.08	0.00	79	2452.0	3482.2	0.06	0.01	103	2399.8	5264.4	0.05	0.03	139
	SIRP-TS	2201.2	2019.2	0.07	0.01	71	2581.2	1679.3	0.08	0.02	81	2828.6	3615.1	0.04	0.10	104
n10m5k3T5	SIRP	4883.5	7056.7	0.09	0.09	180	5099.8	8583.6	0.04	0.03	203	5588.2	10188.8	0.02	0.03	258
	SIRP-T	4094.7	6412.7	0.09	0.00	90	4888.8	7688.5	0.04	0.04	116	6039.0	8406.0	0.09	0.01	156
	SIRPS	4311.4	6434.8	0.01	0.05	96	5176.6	5912.2	0.09	0.07	115	6512.9	7023.6	0.06	0.01	153
	SIRP-TS	4188.9	5257.9	0.02	0.02	89	5040.5	5640.0	0.07	0.08	114	5184.1	5980.0	0.08	0.06	153
n10m5k3T7	SIRP	5518.0	8522.7	0.02	0.01	133	6236.3	9882.4	0.08	0.02	166	8595.4	11407.9	0.07	0.05	213
	SIRP-T	5221.5	7415.2	0.09	0.09	130	6147.2	8751.4	0.04	0.10	168	5789.0	12538.1	0.06	0.04	225
	SIRPS	4863.5	7632.7	0.05	0.06	100	6102.8	8192.9	0.05	0.03	110	5605.6	10562.8	0.07	0.01	147
	SIRP-TS	4904.3	5786.0	0.02	0.01	135	6215.0	6329.8	0.01	0.00	169	7031.5	7160.4	0.03	0.07	209

Table 3.3: Breakdown of cost for a number of product varying between 1 and 5 - number of customers equal to 10

Instances	Model	PD						SPD						NBD					
		T	I	LS	S	Ts	CS (%)	T	I	LS	S	Ts	CS (%)	T	I	LS	S	Ts	CS (%)
n10m1k3T3	SIRP	1590	1340	1173	0	0		2117	2029	893	0	0		2409	2442	1427	0	0	
	SIRP-T	1437	739	541	0	812	14	1696	835	624	0	998	18	2160	1677	192	0	1162	17
n10m1k3T5	SIRP	3430	3325	1775	0	0		3720	4314	1511	0	0		3870	5474	2483	0	0	
	SIRP-T	2995	1819	1077	0	1615	12	3105	2328	1159	0	1852	12	3480	3465	596	0	2213	18
n10m1k3T7	SIRP	4820	4284	3007	0	0		5742	5465	3497	0	0		7044	5540	4369	0	0	
	SIRP-T	4256	2831	1350	0	2222	12	4483	2566	1997	0	2827	19	5655	4828	645	0	2645	19
n10m3k3T3	SIRP	1928	1376	339	1007	0		2219	2188	1243	0	0		2507	2189	1576	0	0	
	SIRP-T	1572	1084	534	0	809	14	1914	1184	596	0	1117	15	2385	1832	291	0	1257	8
	SIRPS	1618	506	490	572	0	32	2144	762	612	571	0	28	1956	1923	605	1247	0	9
	SIRP-TS	1686	182	182	360	418	39	2159	222	96	315	273	46	2605	507	10	532	661	31
n10m3k3T5	SIRP	3768	2982	2853	0	0		4590	4311	2835	0	0		5678	5513	3560	0	0	
	SIRP-T	3401	2167	1134	0	1653	13	4112	2673	1540	0	1819	14	4501	4605	10	0	2935	18
	SIRPS	3538	2385	1067	1557	0	11	3508	2478	1682	2042	0	17	4657	4629	725	2135	0	18
	SIRP-TS	3463	842	526	1598	1925	13	3677	848	305	1633	1728	30	4276	1831	45	2081	1978	31
n10m3k3T7	SIRP	5344	4217	3729	0	0		6190	7214	2757	0	0		7174	8738	2544	0	0	
	SIRP-T	4649	2260	2219	0	2833	10	5194	3327	1872	0	3290	15	5752	5842	462	0	3612	15
	SIRPS	4655	2404	2023	2480	0	13	5397	2952	1906	2660	0	20	5725	6953	10	2591	0	17
	SIRP-TS	4519	1339	1164	2153	2253	14	5264	1184	594	1944	1948	32	5486	2320	46	2892	3278	24
n10m5k3T3	SIRP	2427	2432	1231	0	0		2911	2534	1535	0	0		3107	2532	2478	0	0	
	SIRP-T	2213	1125	1075	0	1068	10	2655	1432	1041	0	1466	6	2873	2444	682	0	1744	5
	SIRPS	2010	1421	859	948	0	14	2452	1395	1004	1084	0	15	2400	3320	1	1944	0	6
	SIRP-TS	2201	381	170	672	796	31	2581	407	190	566	517	39	2829	971	2	1318	1324	21
n10m5k3T5	SIRP	4884	4156	2900	0	0		5100	5575	3009	0	0		5588	5458	4731	0	0	
	SIRP-T	4095	2753	1511	0	2149	12	4889	2701	1841	0	3146	8	6039	4311	875	0	3219	8
	SIRPS	4311	2943	1508	1983	0	10	5177	2529	1627	1757	0	19	6513	4832	211	1981	0	14
	SIRP-TS	4189	683	673	1904	1997	21	5040	1247	434	1857	2101	22	5184	1769	351	1970	1890	29
n10m5k3T7	SIRP	5518	5922	2601	0	0		6236	5506	4376	0	0		8595	7830	3578	0	0	
	SIRP-T	5221	2853	2027	0	2535	10	6147	3830	1910	0	3011	8	5789	8263	381	0	3894	8
	SIRPS	4864	3457	1521	2655	0	11	6103	3981	1835	2377	0	11	5606	6178	852	3533	0	19
	SIRP-TS	4904	817	754	1916	2299	24	6215	1237	654	2396	2043	22	7031	1741	54	2581	2784	29

From Table 3.2 and Appendix B, we can notice that, for all instances and demand patterns under study, any cost reduction is made possible by allowing LT between customers and substituting spare parts. Indeed, from Table 3.3 and Appendix B, we can see that these emergency measures allow for reducing holding and transportation costs along with lost sales.

For all sets of instances and models, we can see that when substitution and LT are not considered, the supply chain seems to experience high transportation, inventory and lost sales costs for the three different distributions. When LT is allowed (SIRP-T), the customers receiving the quantity latterly transshipped are permitted to satisfy even more (sometimes wholly) demand and reduce lost sales. The customers from which the LT is carried out are allowed to lower their inventory holding costs. When the substitution is also allowed (SIRPS), compared to the SIRP model, we observe a reduction in lost sales costs and holding inventory. Indeed, quantities that might be later transshipped from a customer can be used at its level as a substitute for other spare parts, reducing the lost sales. For SIRP-TS, we observe that it considerably reduces costs compared to the other models. Indeed, in addition to what can be received through LT, each customer can use the quantities of spare parts, if compatible, that could constitute idle stock (which leads to high holding cost) to meet the demand for other spares. Moreover, based on our assumptions, PP can also be transshipped to be used as a substitute for additional spare parts using substitution.

Concerning the demand patterns, we observe that a very low variability in-demand size (as in the case of PD) can be less stressful, regardless of the average inter-demand interval. Indeed, when the demand size exceeds the quantity available to be promised, the emergency measures cannot be sufficient to mitigate any loss of sales. Therefore, the benefits of promoting transshipment and substitution rely on the extent to which demand variability is considered a stressful scenario. The computational experiments also stress that the benefit of transshipment and substitution can be less notable for some instances, as in the case of instances n10m1k3T3 and n10m20k3T7 in Table 3.3 with a saving equal to 10% for SIRP-T model. As LT depends on the travelling distances and is incorporated in route decisions, the related cost cannot sometimes be offset by the savings it brings in reducing inventory and lost sales costs. In such context, the substitution would be given higher priority as compared with LT as we could substitute spares at the level of the customer itself, and thus no further shipment would be required. Furthermore, compared with the other configurations, the combination of LT and substitution always allows for a reduction in total costs. On the other hand, the two alternatives help mitigate lost sales, but they can lead to higher costs. Thus, LT and substitution can be of much interest as long as the costs they incur can be offset by the savings they enable.

As for SAA performance, from Table 3.2 for example, we can notice a low variability in the solutions of the SAA runs (based on the standard deviation computed for the UB and LB) which shows the sampling stability of the different runs. We note, however, that the maximum time limit of each SAA run (the 1200s) is reached for some instances and models under consideration (see Tables in Appendix B, for example), which means that some of the SAA runs might not have been resolved to optimality, hindering the quality of the candidate solutions and, in turn, the optimality gap. However, there is no incentive to increase the number of scenarios retained in the SAA problem due to the low variability in the solutions across various instances under the current setting. Moreover, CPU time for some instances is small, meaning optimality is reached (see Table 3.2). For future studies, algorithms can be developed to enhance the quality of solutions within a reasonable amount of time because of the combinatorial complexity of the problem.

3.6 . Conclusions & Perspectives

In this paper, we consider a two-level supply chain. At the first level, a manufacturer-owned central warehouse distributes spare parts to a given number of customers (the second level). Spare parts demand arises when a component fails or requires replacement instead of generated according to end-consumer buying behaviours. We model the problem as a shared inventory-routing problem considering the two flexible instruments of transshipment and substitution to mitigate shortages. We assume that lost sales are allowed when a shortage occurs.

Based on empirical goodness-of-fit tests, three discrete distributions are chosen since they provide a better fit for intermittent demands than continuous ones. These distributions are the Poisson distribution for demand occurrence, combined with demands of constant size, the stuttering Poisson distribution, and the negative binomial distribution.

We have used the SAA method to solve the problem because of its good convergence properties. For the different demand patterns under consideration, computational results highlight that allowing transshipment and substitution is beneficial as they reduce holding and transportation costs along with lost sales. In addition, experiments show the impact of transshipment and substitution on the overall performance depends on the size variability of demands, regardless of the average inter-demand interval. Moreover, they stress that transshipment and substitution can only be of such interest as long as the costs they incur can be offset by the savings they enable. Due to combinatorial complexity, metaheuristics must be developed for future studies to enhance the quality of the candidate solutions. It is also possible to extend this paper to examine a multi-echelon of centralised or decentralised

supply chains. Other policies and non-parametric methods for demand may be investigated, whereby the distribution of empirical distribution is instead directly constructed from the data. It is also possible to investigate stochastic lead time, production, and demands. Moreover, the reliability of PP can be integrated into the model as it affects the customers' future requirements concerning the spare parts to be supplied with. Finally, using metaheuristics and solving the problem in the case of large experimental data sets are necessary to strengthen the present analysis and generalise findings that can be applied to more complex supply chains.

4 - Modeling and solving a dynamic and stochastic multi-product multi-vehicle IRP-T with substitutions

Full Paper published in International Journal Production Research, ISSN 00207543, 1366588X, Impact Factor: 8.6, Quartile: Q1. <https://doi.org/10.1080/00207543.2021.1987549>

Achamrah, F.E., Riane, F. and Limbourg, S. (2021). Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning. International Journal Production Research. pp.1-18.

Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning

Fatima Ezzahra Achamrah ^{a,b}, Fouad Riane ^{a,b}, Sabine Limbourg ^c

^aComplex Systems and Interactions, Ecole Centrale of Casablanca; ^bLaboratoire Genie Industriel, CentraleSupélec, Paris Saclay University; ^cHEC-University of Liege (ULiege)

Abstract: In this paper, we investigate a two-level supply chain consisting of a company which manufactures a set of products and distributes them via its central warehouse to a set of customers. The problem is modelled as a dynamic and stochastic inventory routing problem (DSIRP) that considers two flexible instruments of transshipment and substitution to mitigate shortages at the customer level. A new resolution approach, based on the hybridisation of mathematical modelling, Genetic Algorithm and Deep Reinforcement Learning is proposed to handle the combinatorial complexity of the problem at hand. Tested on the 150 most commonly used benchmark instances for single-vehicle-product DSIRP, the proposed algorithm outperforms the current best results in the literature for medium and large instances. Moreover, 450 additional instances for multi-products DSIRP are generated. Different demand distributions are examined in these experiments: Normal distribution, Poisson distribution for demand occurrence, combined with demands of constant size, Stuttering Poisson distribution, and Negative Binomial distribution. Regarding managerial insights, results show the advantages of promoting inventory sharing and substitutions on the overall supply chain performance.

Keywords: dynamic and stochastic routing, transshipment, substitution, genetic algorithm, deep reinforcement learning

4.1 . Introduction

Highly competitive markets drive companies to efficiently and accurately satisfy their customers' demands across their supply chain. The lead times in most industries must be relatively short, and companies must be flexible enough to meet highly variable demands. Companies also should efficiently manage their capital assets to guarantee profitability. This highly depends on their capacity to maintain their manufacturing and logistical capabilities to meet their customers' service requirements. In practice, to mitigate this

issue, firms may promote inventory sharing among multiple locations within the same distribution network, leading to significant cost reductions. This type of inventory sharing is commonly referred to as lateral transshipment (LT) (Paterson et al., 2011; Grahovac and Chakravarty, 2001).

In such competitive markets, customers choose from various products according to their needs. They may choose to buy their preferred products or replace them with different ones in case of unavailability. Substitutes can lead to healthy market competition between products, which is in the customers' best interest, preventing a market monopoly. This can be the case of food products; perishable products (e.g., artificial blood that can be used as a substitute to mitigate the risks of blood transfusions and shortage of supply or two different milk brands that can be substituted if their "milk" products have similar characteristics); and spare parts (e.g., original equipment manufacturer parts that can be substituted by aftermarket parts called replacement or pattern parts). Substitution could, therefore, serve as a new alternative to better meet customers' demands, mainly if decision-makers are not fully aware of future events. This paper aims to highlight the benefits of promoting both inventory sharing among customers and using substitutes to remedy the shortage of products in such a stochastic environment. Products are therefore considered virtually pooled in the network and sent to a requesting location via LT from a location possessing a surplus of on-hand inventory, or they are substituted, if compatible, by each other.

This paper has four main contributions. First, we study a two-level supply chain in which a manufacturer supplies a central warehouse with a set of products. The central warehouse, distantly located from the manufacturer, distributes products under dynamic and stochastic demands to a given number of customers. Along with LT, substitutions of products, new to literature, are used to sidestep shortages at the customer level. We also assume that direct shipment, if necessary, can take place from the central warehouse to any customer. Secondly, we model the problem as a multi-product dynamic and stochastic inventory routing problem. The objective is to minimise the total cost, including holding inventory, transportation, transshipment, substitution and lost sales. Thirdly, a new resolution approach based on the hybridisation of mathematical modelling, Genetic Algorithm and Deep Reinforcement Learning is proposed to handle the combinatorial complexity of the problem at hand. And finally, tested on the 150 most commonly used benchmark instances for single-vehicle-product DSIRP, our algorithm outperforms the state-of-the-art algorithm. The experimental results show that the proposed algorithm outperforms the current best results in the literature for medium and large instances in terms of the quality of the solutions and run times. In addition, 450 additional instances for multi-product DSIRP are generated. Different demand distributions are examined in these experiments:

Normal distribution, Poisson distribution for demand occurrence, combined with demands of constant size; Stuttering Poisson distribution and Negative Binomial distribution. Results confirm the efficiency of the proposed algorithm and highlight the benefits of both LT and substitutions on the supply chains overall.

The remainder of the paper is structured as follows. Section 2 presents related works. After describing the problem in Section 3, a mathematical formulation is provided in Section 4. In Section 5, a metaheuristic based on hybridisation of mathematical modelling, a Genetic Algorithm and Deep Reinforcement learning is described. Section 6 provides computational experiments. We present conclusions and perspectives in Section 7.

4.2 . Related work

First, we describe the Inventory Routing Problem (IRP) and its classifications, then papers on IRP with uncertainty are categorised, and finally, papers addressing IRP with transshipment are discussed.

4.2.1 . Inventory Routing Problem

IRP includes inventory management, vehicle routing problem (VRP), and delivery scheduling decision-making problems (Coelho and Laporte, 2014). Suppliers can reduce the overall costs of their activities to achieve a competitive advantage by integrating their routing, inventory and distribution decisions instead of independently optimising them. Such decisions can be streamlined by introducing a vendor-managed inventory (VMI) approach, which incorporates replenishment and distribution processes, resulting in overall logistics cost reduction. In Coelho et al. (2012a), IRP is classified according to:

1. the number of customers and suppliers:
 - (a) one-to-one if only one supplier serves one customer (Dror and Levy, 1986).
 - (b) one-to-many, in the most common cases of one supplier and several customers (Bell et al., 1983; Burns et al., 1985; Abdelmaguid, 2004).
 - (c) many-to-many, which occurs less often, with multiple suppliers and multiple customers (Christiansen, 1999; Ronen, 2002).
2. routing can be direct if there is only one client per route, multiple if there are multiple clients on the same route (Zhao et al., 2008), or continuous, as in several maritime applications, where there is no central depot (Savelsbergh and Song, 2008; Hewitt et al., 2013).

3. pre-established inventory strategies to satisfy customers. The two most popular are the Maximum Level (ML) policy and Order-Up to level (OU). The replenishment level is flexible under an ML inventory strategy but is restricted by the resources available to each customer (Coelho and Laporte, 2013a). Under an OU policy, the quantity delivered is required to fill its inventory capacity whenever a customer is visited (Archetti et al., 2007). If the inventory is allowed to become negative, back-ordering will take place, and the corresponding demand will be served at a later period (Abdelmaguid et al., 2009). If there is no back-order, the extra demand will be considered a loss of sales (Mirzaei and Seifi, 2015). In both cases, a penalty for the shortage can be applied.
4. composition and size of the fleet. The fleet can be homogeneous or heterogeneous, and the number of available vehicles can be set at one, set at many or unconstrained (Zhao et al., 2008; Coelho et al., 2012a).

In these papers, only one product is considered, whereas many VMI applications are concerned with multiple product distributions. Few papers address the multi-product inventory routing problem (MPIRP) (Coelho and Laporte, 2013a). Most of the applications emerge in maritime logistics: Bertazzi and Speranza (2002); Grønhaug et al. (2010); Christiansen et al. (2011); Stålhane et al. (2012). Non-maritime cases include, for example, the delivery of perishable goods (Dehghani et al., 2021; Hssini et al., 2016), the transportation of gas by tanker trucks (Bell et al., 1983), the production and the distribution planning in the gas filling industry (Strack et al., 2011), and the vehicle parts industry (Alegre et al., 2007).

4.2.2 . IRP with uncertainty

IRP can be classified into four categories depending on the nature of the input data: (1) static and deterministic; (2) dynamic and deterministic; (3) static and stochastic and (4) dynamic and stochastic. Dynamic IRP (DIRP) differs from the static IRP (SIRP) in that the demands are known before planning in SIRP, while in DIRP, demands are gradually revealed over time (Bertazzi et al., 2013). Stochastic and static IRP (SSIRP) is similar to the static IRP except that the customer demand is known in a probabilistic sense (Bertazzi et al., 2013). In a dynamic and stochastic IRP (DSIRP), the objective is not to deliver a static result, but a solution policy using the information revealed, outlining which measures need to be performed as time passes (Coelho et al., 2014a).

According to Coelho et al. (2014b), solving stochastic DSIRP relies on finding a solution policy which consists of one of the following:

1. optimising a static instance whenever new information becomes avail-

able.

2. applying a static algorithm only once and then re-optimising the problem through a heuristic whenever new information is available.
3. taking advantage of the probabilistic knowledge of future information and using forecasts.

Yu et al. (2012) solve an SSIRP with split delivery using a hybrid approach based on Lagrangian relaxation and local search improvement. Solyali et al. (2012) solve a single product SSIRP with backorders. The authors apply a branch-and-cut algorithm for the robust proposed formulation. Bertazzi et al. (2013) address the same problem under an OU policy and consider shortage to be allowed. The authors present a dynamic programming formulation, a hybrid algorithm based on the roll-out algorithm, and a heuristic method. Huang and Lin (2010) solve a multi-product SSIRP using the conventional ant colony optimisation algorithm. Coelho et al. (2014a) propose an adaptive large neighbourhood search with reactive and proactive policies to solve a single-vehicle single product DSIRP with transshipment. Finally, Roldán et al. (2016) extend the work of Coelho et al. (2014a) by addressing the robustness of inventory replenishment and customer selection policies.

4.2.3 . IRP with transshipment

Coelho et al. (2012b) are, to the best of our knowledge, the first authors to propose the concept of transshipment within inventory routing (IRP-T). The authors propose a mixed-integer linear program to model a single-vehicle and single-product IRP-T. Transshipment is allowed either from the manufacturer to customers or between customers. Lefever et al. (2018) model the same problem as in Coelho et al. (2012b) and strengthen their formulation by proposing a set of valid inequalities for IRP-T based on the existing valid inequalities for the IRP, bounds, reformulation and variable eliminations on the linear relaxation of the problem of concern. Peres et al. (2017) model a multi-period, multi-product IRP-T and use a Randomised Variable Neighbourhood Descent to solve the problem. Hssini et al. (2016) address MPIRP under a static and deterministic demand in a blood supply chain. The authors consider transshipment of blood products between hospitals and substitution between blood groups. On stochastic demand, Dehghani et al. (2021) develop a mathematical model that decides on transshipment under static and stochastic demand to reduce total costs and shortages in a blood supply chain. Achamrah et al. (2022c) model a two-level spare parts supply chain under static and stochastic demands. The authors consider transshipment of spare parts between depots and substitutions between original equipment manufacturer and pattern parts. Chrysochoou et al. (2015) propose a two-stage programming model in which transshipment is considered a recourse

action to address a single product and vehicle DSIRP. [Coelho et al. \(2014a\)](#) address the dynamic and stochastic version of the problem studied by [Coelho et al. \(2012b\)](#). Under OU and ML policies, the problem is solved using either a proactive or reactive policy, all implemented in a rolling horizon fashion. In the proactive policy, once forecasts on demands are obtained, routes are constructed, and LT takes place after the demand is realized to reduce shortages. The reactive policy (or wait and see policy) observes the system's state and makes decisions accordingly. It is defined as an (r, S) replenishment system under which whenever the inventory reaches the reorder point r , it triggers a replenishment order to bring the inventory position up to S . Routing is constructed based on the threshold r , and as in proactive policy, LT occurs when demands are revealed. Authors also use an adaptive large neighbourhood search to determine routing and the exact method to determine the quantities to be transshipped. The setting of our problem description follows that of this paper.

4.2.4 . Paper main contributions

Based on this literature review, apart from promoting transshipment between customers to avoid shortages of products, none of the existing papers incorporates product substitution within a dynamic and stochastic setting. The present paper extends the work of [Coelho et al. \(2014a\)](#) on a single product and single-vehicle DSIRP by addressing a more realistic configuration of the problem at hand. In the following, we study a one-to-many multi-product DSIRP under an ML policy in which customers' demand follows a probability distribution in which values of parameters are revealed over time. Moreover, we propose a model that integrates substitutions along with transshipment as alternatives to sidestep shortages at the customer level. As for the resolution approach, the present study also contributes to the existing literature by combining Genetic Algorithm and Deep Reinforcement Learning techniques. The latter is used to analyze data related to the decision and the objective spaces visited during the search process, moves and recombination. With the help of Deep Q-learning, helpful knowledge is extracted and used to enhance the search performance and speed of the metaheuristic. Applied on a benchmark of 150 instances with up to a maximum of 200 customers and 20 as the number of periods for a single-product DSIRP and on a set of 450 generated instances for multi-products, the resolution approach obtains results that are advantageous compared to results stemming from the state-of-the-art algorithm, thus, proving its efficiency.

4.3 . Problem setting

The following problem description follows that of the paper of [Coelho et al. \(2014a\)](#) developed for a single-vehicle-product DSIRP. Our multi-vehicle-product DSIRP with Transshipment and Substitution (DSIRP-TS) is defined on a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where \mathcal{N} is the vertex set indexed by $i \in \{0, \dots, n\}$ and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$ is the edge set. Vertex 0 represents the the manufacturer's central warehouse (CW), and the set $\mathcal{N}_0 = \mathcal{N} \setminus \{0\}$ denotes the customers. The planning horizon length is T with discrete time periods $t \in \mathcal{H} = \{1, \dots, T\}$. The demand d_{pit} each customer $i \in \mathcal{N}_0$ has to satisfy for product $p \in \mathcal{P} = \{1, \dots, m\}$ per period $t \in \mathcal{H}$ is a random variable D_{pit} per stock keeping unit (SKU). Moreover, each customer and the central warehouse, $i \in \mathcal{N}$, incur unit inventory holding costs, h_{pi} per product $p \in \mathcal{P}$, with inventory capacities K_i . Inventories are not allowed to exceed the holding capacity and must be positive. We further assume the CW has enough inventory to meet all demand during the planning horizon. At the beginning of each period, at each location $i \in \mathcal{N}$, the current inventory levels I_{pi0} of the product p are known.

A set of homogeneous vehicles $v \in \{1, \dots, k\}$ is available, each with a capacity Q in terms of SKU with routing cost c_{ij} associated to all $(i, j) \in \mathcal{A}$. Direct deliveries and multiple routing are permitted to guarantee that all planned deliveries are met (before demands are revealed). LT can occur when it is profitable to ship products between customers. LT between customers and direct shipment from CW to any customer are assumed to be outsourced, and the relative unit cost can be expressed as αc_{ij} , where $\alpha > 0$. α is used to express that outsourced operations are volume-dependent rather than distance-dependent (as this is how often carriers define the terms of contracts).

The unit cost of substituting a product p by $s \in \mathcal{P}$ is a_{ps} . All possible combinations according to the products' compatibility are represented by o_{ps} , which is equal to 1 if a product p is compatible, according to the customer, with a product s , and 0 otherwise. Compatible products can be used as substitutes to satisfy customer demand when preferred products are unavailable. We assume that the substitution of products is not bi-directional. A product p is substituting product s , but the inverse is not necessarily implied. We also assume that the CW distributes multiple products, including substitutes, and deliveries and transshipment can be performed simultaneously. The lost sales cost, which is associated with the shortage of a product p at the customer i is f_{pi} . Finally, we assume that the manufacturer has enough inventory of products to service its CW and that the quantity of product p shipped from the manufacturer to the CW in period t is expressed by g_{pt} . As in [Coelho et al. \(2014a\)](#) and [Archetti et al. \(2012\)](#), we assume that g_{pt} is used only to account for inventory costs at the CW.

Regarding the sequence of the operations, we assume that the decisions

related to routing, including direct shipments, are determined. Second, after demands are realised, LT and possible substitutions are performed to sidestep, as much as possible, shortages at the level of each customer. Decisions variables are as follows:

- I_{pit} the inventory level of product p at node $i \in \mathcal{N}$ at the end of a period t .
- q_{piv} the quantity of product p delivered by vehicle v from the CW to the node $i \in \mathcal{N}$ in a period t .
- w_{pijt} the quantities of product p carried by the outsourced carrier from the the node $i \in \mathcal{N}$ to $j \in \mathcal{N}$, in a period t .
- l_{pit} the lost sales quantity of product p at customer $i \in \mathcal{N}_0$ in a period t .
- z_{spi} are defined for all $o_{sp} = 1$ as the quantity of a product s substitute for product p used at the customer i in a period t to satisfy a part of the unsatisfied demand.

The inventory level at the end of each period at customer i is then:

$$I_{pit} = I_{pit-1} + l_{pit} + \sum_{j \neq i \in \mathcal{N}_0} (w_{pjit} - w_{pijt}) + \sum_{s \neq p \in \mathcal{P}} (z_{spit} - z_{psit})$$

$$\forall p \in \mathcal{P}, i \in \mathcal{N}_0, t \in \mathcal{H} \quad (4.1)$$

The objective function is to minimise the total cost which includes inventory holding, lost sales, substitutions, transshipment and routing costs:

$$\min \sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}} \sum_{p \in \mathcal{P}} h_{pi} I_{pit} + \sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}_0} \sum_{p \in \mathcal{P}} f_{pi} l_{pit} +$$

$$\sum_{t \in \mathcal{H}} \sum_{i \in \mathcal{N}_0} \sum_{p, s \in \mathcal{P}} a_{sp} z_{spit} + \alpha \sum_{t \in \mathcal{H}} \sum_{i, j \in \mathcal{N}_0, i \neq j} c_{ij} \sum_{p \in \mathcal{P}} w_{pijt} + \mathcal{C}_t \quad (4.2)$$

where \mathcal{C}_t is the cost of the routes performed in a period t . In the following, we present how the DSIRP-T is solved using a reactive policy as in (Coelho et al., 2014a).

4.4 . Solution policy

The reactive policy, also known as the "wait and see" policy, consists of observing the system's state to make decisions regarding routing, transshipment and substitution. As in (Coelho et al., 2014a) we adopt an (r_{pit}, S_{pit}) replenishment system in which anytime the inventory level of a product p

reaches a reorder point r_{pit} , a replenishment order to visit a customer i is triggered so that the inventory level is brought up to a value S_{pit} . Routing, including direct shipment, is constructed accordingly (Routing Model RM) in each period t . The objective is to deliver the quantities of products that have been determined using RM. Then, when visiting customers, the product p 's demand is revealed in a period t and LT, and substitution occurs if the available inventory is insufficient to meet this demand (Transshipment and Substitution Model TSM). This is implemented in a rolling horizon framework. Figure 4.1 provides a numerical example. Notations of the model are summarised in Table 4.1.

Table 4.1: Notation summary

Sets	
H	Planning horizon indexed by t
V	Set of CW's vehicles indexed by v
\mathcal{P}	Set of products indexed by p
Routing model	
Parameters	
c_{ij}	Transportation unit cost associated to regular shipment using CW's vehicles for all $(i, j) \in \mathcal{A}$
α	Discount factor $0 < \alpha < 1$ used to represent the cost associated to the outsourced operations
Q	CW's vehicles capacity
I_{p00}	Inventory level at beginning of the planning horizon of a product p at CW (node 0)
g_{pt}	Quantity of a product p supplied to CW in period t
r_{pit}	Reorder point for product p , at customer i and period t
u_{pit}	Expected demand of product p in period t
σ_{pit}	Standard deviation of the demand of product p in period t
β	Shortage probability
$\Upsilon_{\beta pt}$	β -order quantile of the demand distribution for product p
\mathcal{I}_{pit-1}	Inventory level of product p at customer i at the beginning of period t (before the demand is revealed)
d'_{pit}	Quantity of product p that should be delivered to customer i in period t (using both regular and direct shipment/ before the demand is revealed)
Decision variables	
q_{pitv}	Quantity of product p delivered by vehicle v of CW to node $i \in \mathcal{N}$ in period t
w_{p0it}	Quantity of product p carried by the outsourced carrier from CW to node $i \in \mathcal{N}_0$, in period t
I_{p0t}	Inventory level of a product p at CW (node $\{0\}$) at the end of period t
x_{ijv}	Equal to 1 if the arc $(i, j) \in \mathcal{A}$ is visited by vehicle v in period t ; 0 otherwise
y_{itv}	Equal to 1 if a customer i is visited by vehicle v in period t ; 0 otherwise
Transshipment and substitution model	
Parameters	
h_{pi}	Unit inventory holding cost of product p at customer $i \in \mathcal{N}_0$ in period t
f_{pi}	Unit cost associated to the lost demand of product p at customer $i \in \mathcal{N}_0$ in period t
a_{sp}	Unit cost associated with the substitution of a primary preferred product p by a substitute s
D_{pit}	Random variable associated to revealed demand of product p in period t at node $i \in \mathcal{N}_0$
K_i	Maximum inventory capacity at customer $i \in \mathcal{N}_0$
I_{pi0}	Inventory level of product p at customer $i \in \mathcal{N}_0$ at beginning of the planning horizon
o_{sp}	Equal to 1 if a substitute product s is compatible with a primary preferred product p , and 0 otherwise
Decision variables	
I_{pit}	Inventory level of a product p at a customer i after the demand is revealed and the performance of transshipment and substitutions at the end of period t
w_{pijt}	Transshipment quantity of product p carried by the outsourced carrier from customer $i \in \mathcal{N}_0$ to the customer $j \in \mathcal{N}_0$, in period t after the demand is revealed
z_{spit}	Quantity of product s substitute for product p used at the customer i in period t to satisfy a part of the unsatisfied demand (defined for all $o_{sp} = 1$)
l_{pit}	Lost sales quantity of product p at customer $i \in \mathcal{N}_0$ in period t when the demand is revealed and performance of transshipment and substitutions

Note that the part of the satisfied demand of product p is represented by the quantity z_{ppi} of the product p used as a substitute of itself.

4.4.1 . Routing model

Under ML policy, CW can freely decide on the quantity to supply the customer with, restricted only by the customer's inventory capacity and the threshold r_{pit} . This quantity defines the parameter d'_{pit} that is proportional to $\max[0, r_{pit} - \mathcal{I}_{pit-1}]$, where \mathcal{I}_{pit-1} is the inventory level at the beginning

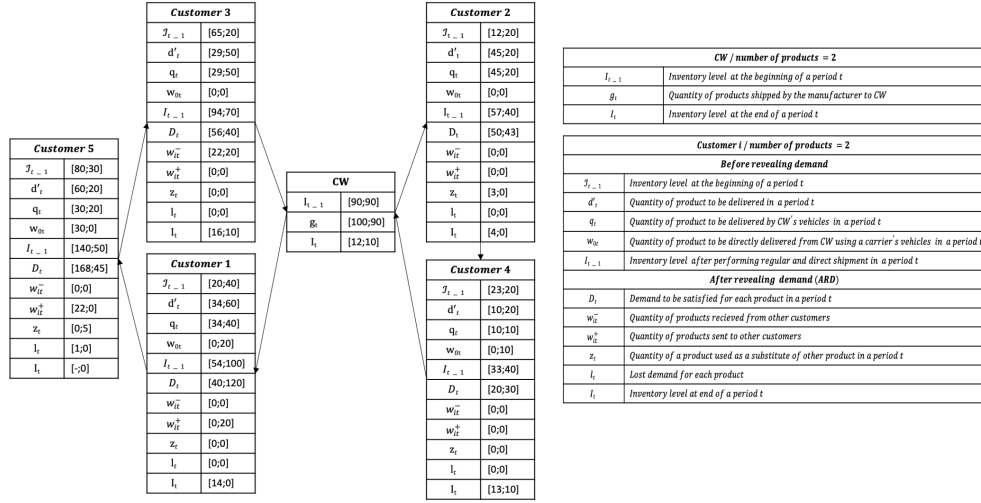


Figure 4.1: A numerical example.

of a period t . r_{pit} is defined as the expected demand during a lead time \mathcal{L} (which is equal to 1 as deliveries taking place in a period t can be used to satisfy demand at period t), plus a safety stock which depends on demand variability, \mathcal{L} and target service level. We assume a normally distributed demand as in [Coelho et al. \(2014a\)](#). r_{pit} can be then computed as follows:

$$r_{pit} = u_{pit} + \Upsilon_{\beta pt} \sigma_{pit} \quad (4.3)$$

where u_{pit} in a given period t is an estimate of the expected demand of product p at the customer i and σ_{pit} the related standard deviation. β is the shortage probability and $\Upsilon_{\beta pt}$ is the β -order quantile of the demand distribution for the product p . These values as well as r_{pit} are updated in each period t .

To construct vehicle routing in each period t , a mixed-integer linear program (MILP) is proposed. The objective is to decide which customer is allocated to which vehicle, the quantities delivered at each node, and direct shipments if any.

For each period t , the RM is formulated as follows :

Objective function OF of routes X :

$$\min \left[\sum_{v \in V} \sum_{i, j \in \mathcal{N}, i \neq j} c_{ij} x_{ijvt} + \alpha \sum_{i \in \mathcal{N}_0} c_{0i} \sum_{p \in \mathcal{P}} w_{p0it} \right] \quad (4.4)$$

Subject to:

$$\sum_{v \in V} q_{pitv} + w_{p0it} = d'_{pit} \quad \forall p \in \mathcal{P}, i \in \mathcal{N}_0 \quad (4.5)$$

$$\sum_{i \in \mathcal{N}_0} \sum_{p \in \mathcal{P}} q_{piv} \leq Q \quad \forall v \in V \quad (4.6)$$

$$\sum_{p \in \mathcal{P}} q_{piv} \leq Q y_{itv} \quad \forall i \in \mathcal{N}_0, v \in V \quad (4.7)$$

$$\sum_{j \in \mathcal{N}, i \neq j} x_{ijtv} + \sum_{j \in \mathcal{N}, i \neq j} x_{jiv} = 2y_{itv} \quad \forall i \in \mathcal{N}, v \in V \quad (4.8)$$

$$\sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}, i \neq j} x_{ijtv} \leq \sum_{i \in \mathcal{S}} y_{itv} - y_{itv} \quad \forall \mathcal{S} \subseteq \mathcal{N}_0, \mathcal{S} \neq \emptyset, v \in V \quad (4.9)$$

$$I_{p0t} = I_{p0t-1} - \sum_{i \in \mathcal{N}} \sum_{v \in V} q_{piv} - \sum_{i \in \mathcal{N}} w_{p0it} + g_{pt} \quad \forall p \in \mathcal{P} \quad (4.10)$$

$$q_{piv}, w_{p0it} \geq 0 \quad \forall p \in \mathcal{P}, i \in \mathcal{N}_0, v \in V \quad (4.11)$$

$$x_{i0tv} \in \{0, 1, 2\} \quad \forall i \in \mathcal{N}_0, v \in V \quad (4.12)$$

$$x_{ijtv} \in \{0, 1\} \quad \forall i, j \in \mathcal{N}_0, v \in V \quad (4.13)$$

$$y_{itv} \in \{0, 1\} \quad \forall i \in \mathcal{N}_0, v \in V \quad (4.14)$$

The objective function (4.4) is to minimise the cost of routing and direct shipments. Constraints (4.5) defines the total quantity supplied to a given customer d'_{pit} with respect to the delivery modes. Constraints (7.5) ensure that vehicle capacity is not exceeded and constraints (4.7) stipulate that CW's vehicle supplies quantities only to customers allocated to a visit. Constraints (4.8) and (4.9) are respectively degree and sub-tour elimination constraints. The conservation conditions of inventory at the central warehouse are expressed by constraints (4.10). Constraints (4.11)-(4.14) state the conditions of non-negativity and integrality on the variables.

4.4.2 . Transshipment and substitution model

After the routing decisions have been constructed based on the (r_{pit}, S_{pit}) system, in TSM, the objective is, therefore, to use transshipment and substitution as emergency measures whenever demands have been revealed exceeds the quantity of products made available to each customer. In TSM, an inventory level I_{pit-1} refers to the initial inventory per product at the beginning of each period of the rolling horizon. TSM is solved for each period t after

demands have been revealed.

For each period t , TSM is formulated as follows:

$$\min \left[\sum_{i \in \mathcal{N}_0} \sum_{p \in \mathcal{P}} h_{pi} I_{pit} + \sum_{i \in \mathcal{N}_0} \sum_{p \in \mathcal{P}} f_{pi} l_{pit} + \sum_{i \in \mathcal{N}_0} \sum_{(p,s) \in \mathcal{P}} a_{sp} z_{spit} + \alpha \sum_{(i,j) \in \mathcal{N}_0, i \neq j} c_{ij} \sum_{p \in \mathcal{P}} w_{pijt} \right] \quad (4.15)$$

Subject to:

$$I_{pit-1} = \mathcal{I}_{pit-1} + d'_{pit} - D_{pit} \quad \forall p \in \mathcal{P}, i \in \mathcal{N}_0 \quad (4.16)$$

$$I_{pit} = I_{pit-1} + l_{pit} + \sum_{j \neq i \in \mathcal{N}_0} (w_{pjit} - w_{pijt}) + \sum_{s \neq p \in \mathcal{P}} (z_{spit} - z_{psit}) \quad \forall p \in \mathcal{P}, i \in \mathcal{N}_0 \quad (4.17)$$

$$0 \leq I_{pit} \leq K_i \quad \forall p \in \mathcal{P}, i \in \mathcal{N}_0 \quad (4.18)$$

$$I'_{pit-1} = I_{pit-1} + \sum_{s \neq p \in \mathcal{P}} (z_{spit} - z_{psit}) \quad \forall p, s \in \mathcal{P}, i \in \mathcal{N}_0 \quad (4.19)$$

$$0 \leq l_{pit} \leq -\min[0, I'_{pit-1}] \quad \forall p \in \mathcal{P}, i \in \mathcal{N}_0 \quad (4.20)$$

$$0 \leq w_{pijt} \leq \min[\max[0, I'_{pit-1}], -\min[0, I'_{pjt-1}]] \quad \forall p \in \mathcal{P}, i, j \in \mathcal{N}_0 \quad (4.21)$$

The objective function (4.15) is to minimise the cost of inventory holding, lost sales, substitution and transshipment costs. Constraints (4.16) define actual inventory level after demands is revealed. Constraints (4.17) state that the inventory level at the end of the period of a product p at the level of each customer i is computed using the actual inventory level at i , quantities transshipped from and to customer i and the difference between the quantity of product s used as a substitute of p and the quantity of p used as a substitute of the other products. Constraints (4.18) impose bound on inventory level. Constraints (4.19) define the inventory level after products' substitutions have taken place. Constraints (4.20) state that if the initial inventory of a product p is non-negative, both boundaries are equal to zero, and thus $l_{pi} = 0$, i.e. no demand is lost; otherwise, the number of lost units is maximum $-I'_{pit} - 1$. Constraints (4.20) state that if the initial inventory of a product p is non-negative, then no demand is lost, and both boundaries are equal to zero; otherwise, a minimum of zero and a maximum of I'_{pit-1} units are lost. Similarly, for each product p , constraints (4.21) place limits on the transshipment arc flows. For customers i and j , there are four possible combinations of inventory levels, all of which can be managed by these constraints:

- $I'_{pit-1} < 0$ and $I'_{pjt-1} < 0$: no transshipment is possible since there is not enough inventory to ship to j .
- $I'_{pit-1} \geq 0$ and $I'_{pjt-1} < 0$: I'_{pjt-1} is the upper bound on the arc of the emergency transshipment from i to j .
- $I'_{pit-1} < 0$ and $I'_{pjt-1} \geq 0$: no transshipment is needed since customer j does not need LT and i does not have enough inventory.
- $I'_{pit-1} \geq 0$ and $I'_{pjt-1} \geq 0$: no transshipment is needed since customer j has enough inventory.

4.5 . Genetic Algorithm and Deep Reinforcement Learning

The classical Vehicle Routing Problem (VRP) is NP-hard (Laporte, 2009). Consequently, exact methods can fail to find optimal solutions for large-size problems. Given the complexity of the RM, our approach is, therefore, to first use a metaheuristic, namely Genetic Algorithm (GA), to determine routing decisions. Unlike Neighborhood Search Algorithms known for their propensity to deliver only local optima solutions, GA is an efficient computational tool known for its simplicity, great global search ability and adaptable topology (Baker and Ayechev, 2003). However, GA does not quietly scale well with complexity (as explained in Chapter 2). On the other hand, metaheuristics in general and GA in particular, through their iterative search processes, generate a lot of data that can be turned into explicit knowledge if coupled with machine learning models (Talbi, 2020). This data relates to decision-making solutions and the objective spaces visited during the search process, solution or trajectory sequence, successive solution populations, movements, recombination, local optima, elite solutions, bad solutions, etc. Machine learning techniques may assist in analysing this data, learning useful knowledge and guiding to improve metaheuristics' search performance and speed. Thus, techniques for metasearch are *data-driven, well informed* and therefore *smarter*. In this paper, we use Deep Q-learning (DQ) to speed up our GA, which combines reinforcement learning (RL) and deep learning techniques. We further explain these steps in the following sections.

Once RM is solved in the current period, the solution is used as a parameter for TSM, which is solved exactly using CPLEX with default parameters. To do so, we use a matheuristic (noted DQ-GA) in a rolling horizon framework which hybridises the exact method and GA. We now describe these steps in detail.

4.5.1 . Genetic Algorithm

The algorithm begins with a set of initial solutions called population. For each slice time of the rolling horizon, each individual in the population is referred to as a chromosome, reflecting the sequence of assigned customers to each vehicle. During each generation, the fitness of each solution is measured, and solutions are evaluated and selected for cloning, crossover and mutation operations based on their fitness (computed using objective function values).

4.5.1.1 . Chromosomes encoding

In this paper, each chromosome X is represented using a one-dimensional array of integer values, representing the nodes (customers) to be visited (see Figure 4.2). A repair heuristic is used to check the RM constraints. It ensures, for instance, that no customer with a non-zero d'_{pi} is missing on the routes, or it belongs to several routes.

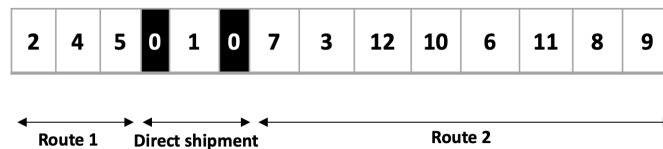


Figure 4.2: Representation of a chromosome for an RM with 12 customers and 2 vehicles.

4.5.1.2 . Generating the initial population

We use a variant of the 2-opt heuristic to generate an initial population for GA, an algorithm based on the conditional permutation of nodes (Sabba and Chikhi, 2012). The heuristic begins by randomly selecting two nodes in a tour and allowing permutation between segments as long as the total cost is reduced. Also, this permutation relies on inter-route moves. That is, we permit swapping nodes that belong to different tours. This process is repeated until routes are optimised.

4.5.1.3 . Fitness function

The fitness function of a chromosome X is calculated from the objective function $OF(X)$ of RM as follows:

$$F(X) = \frac{1}{OF(X)} \quad (4.22)$$

4.5.1.4 . Genetic Operators

In this algorithm, the following operators are used:

- The cloning operator retains the best solutions found so far. The trade-off between the performance of the algorithm and its speed led to picking the best 20% of the present population of chromosomes to be copied into the next generation.
- The parent selection operator uses a binary selection process that begins with two chromosome pairs. Two chromosomes are selected randomly from the existing population each. For crossover operations, the two best chromosomes are selected for each pair. This leads to two children, each counting in the new population.
- The crossover operator is necessary to mate the chromosome pairs so that they produce their offspring. This paper implements double-point crossover to guarantee the preservation of the best chromosomes. A crossover is performed based on a P_C probability.
- The mutation operator is a second operator used to explore new neighbours. It consists of producing random alterations in different chromosomes. A reversal mutation is used since it is shown to be effective (Zhang et al., 2010). A random set of two nodes are selected, and the nodes between are reversely ordered. Like the crossover, the mutation process is performed with a P_M probability. Accordingly, each node in a chromosome is checked for possible mutation by generating a random number between zero and one, and if this number is less than or equal to P_M , the node value is changed.

4.5.1.5 . Constraints violation penalty

A simple penalty strategy is adopted to respect the constraints of the model. In other words, the feasibility of each chromosome is tested in light of the violations of the model constraints during the generation of initial solutions, along with the crossover and mutation operations. If there is an infeasibility in the solution, then the value of the fitness function of the corresponding chromosome is correlated with a penalty. In this way, infeasible chromosomes are less likely to integrate into the next generation of chromosomes.

GA stops when a time limit is reached, or no improvement in the quality of the solution is noted.

4.5.2 . Deep Reinforcement

In this section, we present the deep RL algorithm used to speed up the convergence of GA.

4.5.2.1 . Q-learning

Q-Learning is an RL off-policy algorithm characterised by its strong self-adaptability and environmental feedback signals (Alom et al., 2019). The main idea is to use the feedback signal to adjust an agent's action policy to optimise its choice when interacting with an environment. By performing actions (i.e., genetic operators), the agent (a chromosome here) arrives in different conditions known as states. Actions contribute to rewards that can be positive and negative. The idea behind Q-learning consists of putting the agent in sequences of state-action pairs, observing the resulting rewards, and adjusting the predictions of a table (called a Q-table) to those rewards until correctly predicted by the best policy. The "Q" stands for quality, which measures how beneficial a given action is in achieving a potential reward.

An agent communicates with the environment in one of two ways: exploration and exploitation (Silver, 2015). Exploration consists of allowing the agent to choose the action randomly it will take regardless of the reward, while in exploitation, the agent uses the Q-table and selects an action depending on the maximum reward. Initially, the exploration rate noted ϵ (also called ϵ -greedy policy) is set to 1 as all the actions have a Q-value of 0. As the agent learns more about its environment, ϵ is decayed by a specific rate so that the probability of exploration decreases.

4.5.2.2 . Deep Q-learning

Q-learning is a straightforward and efficient algorithm for our GA to construct a Q-table. This allows the latter to figure out exactly the best actions to perform for crossover, cloning and mutation operators in terms of the best moves. However, it could be time-consuming since the amount of memory needed to save and update the table will increase as the number of states increases, and the amount of time required to explore each state to build the appropriate Q-table would be impractical. Since computational time is our primary concern, we estimate these Q-values with deep learning models, namely neural networks, known as DQ (Adams et al., 2021; Zhang et al., 2010). Indeed, to approximate the Q-value function, we use a neural network. This function maps a state to the Q values of all the actions that can be taken from that state. It learns the network's parameters (weights) to output the optimal Q-values. Choosing the correct action means comparing the possible rewards of each action and selecting the best one.

4.5.2.3 . DQ exploitation

As depicted in Figure 4.3, DQ starts with random Q-value estimations and uses the ϵ -greedy policy to explore the environment. DQ improves its Q-value estimates using the same concept of dual actions, a current action with a current Q-Predicted value and a target action with a Target Q-value. As the network and its weights are identical, the direction of the predicted Q-Target values changes; they remain unchanged but may fluctuate following each update. The stabilisation of the Q-Target values is ensured by using a second network which is not trained. The learned weights from the Q-Predicted Network are copied to the Q-Target network after a pre-set number of steps noted C-iteration. From Figure 4.3, we can see two neural networks in the DQ architecture (Q-Predicted and Q-Target) and an agent Experience Replay. Replay Experience interacts with the environment for data generation for Q-network training. This information contains all moves carried out by GA's operators and saved as $\langle st, a, \mathcal{R}, st' \rangle$ tuples (see notation below Equation 6.53). Then a sample is picked randomly from this data, consisting of a mix of older and more recent samples. This batch of training data is used in the Q-Predicted and Q-Target networks. The Q-Predicted network takes the current state and moves out of each sample, and that particular move predicts the Q value. Q-Predicted value, Q-Target value and the observed data sample reward are used to compute the loss for the

Q network training (see Equation 6.53). To reduce variance and guarantee the stability of the algorithm, in C-iteration, a batch of data is selected from all prior experiences.

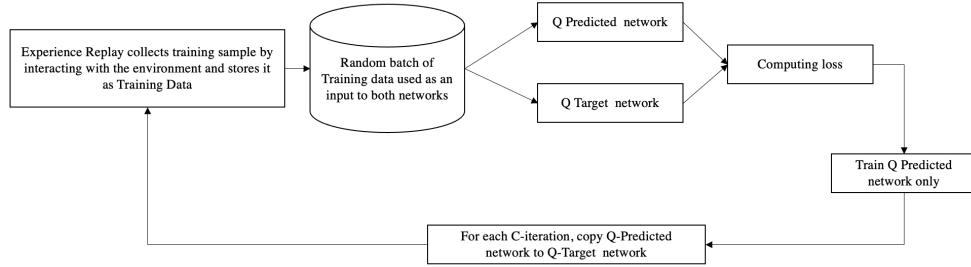


Figure 4.3: DQ architecture

$$Loss = [\mathcal{R}_{t+1} + \gamma \max_a (\theta^T Q(st', a') - \theta^T Q(st, a))]^2 \quad (4.23)$$

Where:

- γ : discount-rate parameter. It measures how much weight the future awards are given.
- a, a' : current and future action respectively.
- st, st' : current and future state respectively.
- \mathcal{R}_{t+1} : future reward.
- $Q(st, a)$: learned action-value function.
- θ^T : Transpose matrix of network weights.

Finally, to further speed up the GA, all genetic moves we have gotten as of yet are stored. Instead of "starting from scratch" every time the algorithm is run to solve the RM either for the current instance, for a different period, or a new given instance, which happens to be similar to the chromosomes that have already been treated, we use the "memory" to exploit the best optimal policies rapidly. The selection of the best moves depends on how the instance is solved, similar to the previously addressed instances. The K-Nearest Neighbours algorithm is used to determine clusters of instances closer to a given "new and unseen" instance (Mohammed et al., 2017).

4.6 . Computational Results

4.6.1 . Experimental design

First, to test the effectiveness and validate the proposed resolution approach, we perform experiments on a set of well-known benchmark instances developed for the single-vehicle and product DSIRP with and without transshipment. It is composed of 150 instances proposed by Coelho et al. (2014a): 5 to 200 customers and a planning horizon of 5 to 20 periods for a total of 10 instances for each set of customers. The instances follow some standards defined for the deterministic IRP instances of Archetti et al. (2012, 2007), namely the mean customer demand, initial inventories, vehicle capacity and distances matrix. The demand follows a normal distribution, the mean is generated as an

integer random number after an interval of discrete uniform distribution [10, 100], and the standard deviation is generated as an integer random number after an interval of discrete uniform distribution [2, 10]. A negative demand value, if generated, is replaced by zero. The maximum inventory capacity is a multiple of the average demand, and initial inventories equal the maximum capacity minus the average demand. In the interval [0.02, 0.10], holding costs are generated randomly from a continuous uniform distribution, and the shortage penalty cost equals 200 times the cost of holding. Finally, the unit cost α is set to 0.01 and the vehicle capacity to $1.5 \sum_{i \in \mathcal{N}_0} \sum_{p \in \mathcal{P}} u_{pi}$. Where u is the expected demand. All instances are available from: <https://www.leandro-coelho.com/instances/inventory-routing/> and solutions are retrieved from [Coelho et al. \(2014a\)](#). Finally, a fair comparison between algorithms and hardware benchmarking is used to compare the algorithms' speed. The reported CPU of the matheuristic is thus recalculated to align the computational time concerning the computer's performance in [Coelho et al. \(2014a\)](#). Further information on the CPU speed of both computers can be found on: www.cpubenchmark.net.

Second, we consider a set of randomly generated instances to evaluate the DSIRP-TS for multi-products and multi-vehicles and highlight the benefit of transshipment and substitution on the supply chain's overall performance. The generation precisely follows the standards defined for the single-vehicle-product DSIRP. As the supplier has a fleet of homogeneous vehicles, the vehicle capacity no longer needs to be expressed as a function of the expected demands. In this paper, we consider a set of 10 homogeneous vehicles, each having a capacity of $Q=2000$ units. For products' substitution, we consider a constant unit cost of $c = 0.1\$/\text{product}$ (for identical products: $s = p$, $c = 0\$/\text{product}$). All optimisation steps were carried out with a personal computer (MacBook Pro, macOS Big Sur, 3.3 GHz Quad-Core Intel Core i7 CPU with 8 GB of RAM) and with CPLEX 12.9 for the resolution of TSM and Python for RM, Python and Pytorch for DQ.

4.6.2 . Parameters tuning

Sophisticated optimisation algorithms typically require a large number of parameters to be set to enhance their performance. The immediate purpose of the automated configuration of the algorithm is to find the optimiser's best parameter settings automatically. Automatic configuration of algorithms can contribute to new design paradigms for optimisation applications. The Irace package is a software package that implements various automated configuration procedures ([López-Ibáñez et al., 2016](#)). It provides particularly iterated racing procedures that have been used effectively to configure various state-of-the-art algorithms automatically. Irace's repeated racing processes include the iterated F-race algorithm and several improvements and extensions. In this paper, a set of training instances representing the problem (40 instances with 5, 10, 15... 50 customers each) is used to choose the best algorithm configuration (see Table 4.2). The selected algorithm configuration can then be used to solve new instances of the same problem.

Table 4.2: Parameters tuning using Irace package

Parameters	Range	Chosen values
Crossover probability P_C	[0.60,0.81]	0.7
Mutation probability P_M	[0.33,0.37]	0.34
Population size	[100;140]	110
Maximum number of iteration	[100;140]	110

4.6.3 . Computational results for the single-vehicle-product DSIRP with and without transshipment

In this section, we present the results of the experiments performed on the set of small to large-scale datasets generated by [Coelho et al. \(2014a\)](#). To assess the performance

of the matheuristic, we compare it with the result obtained using the best known ALNS of [Coelho et al. \(2014a\)](#) developed to solve a single vehicle-product DSIRP. A statistical analysis using ANOVA is also conducted to assess the randomness or not of the differences between the obtained results ($p\text{-value} > 0.05$). For each size of instance (small, medium and large), we present the average total cost and the CPU time in seconds. Results are summarised in Table 4.3.

First, for all 150 instances under consideration, we notice that on average, our algorithm provides better solutions both in terms of CPU and costs than those of [Coelho et al. \(2014a\)](#) apart from small instances, with a very small gap between the costs. Thus, our algorithm is competitive and efficient compared to the most known state-of-the-art algorithm applied to solve a such specific DSIRP-T under a reactive policy. Finally, as is expected, we notice that sharing inventories between customers helps to reduce the lost sales and thus total costs significantly.

Table 4.3: Summary of comparison between the results obtained by [Coelho et al. \(2014a\)](#) and this paper

Instances	Average cost				Average CPU (s)			
	Coelho et al. (2014a)		DQ-GA		Coelho et al. (2014a)		DQ-GA	
	DSIRP	DSIRP-T	DSIRP	DSIRP-T	DSIRP	DSIRP-T	DSIRP	DSIRP-T
Small	10,225.9	7,926.7	9,473.6	8,788.9	46.3	44.6	42.7	11.6
Medium	30,360.7	26,527.1	27,797.7	26,244.4	452.7	444.1	125.2	129.0
Large	61,250.2	54,292.4	50,550.0	47,352.2	3,860.1	4,100.1	136.6	127.0
Average	33,945.6	29,582.0	29,273.8	27,461.8	1,453.0	1,529.6	101.5	89.2

4.6.4 . Computational results for the multi-vehicle-product DSIRP with substitution and transshipment

We now evaluate the impact of transshipment and substitution on solution cost for a more realistic DSIRP. We first consider DSIRP for 20 products and compare the results obtained for DSIRP without transshipment (DSIRP), DSIRP with transshipment (DSIRP-T), and DSIRP with substitution (DSIRPS) and finally DSIRP with transshipment and substitution (DSIRP-TS). Later, we apply the same logic to 40 products. The aim is to confirm the representativeness of the results, highlight the benefits of both transshipment and substitution, and re-evaluate the proposed algorithm's accuracy. The generation of the 300 instances (150 for the 20 and 40 products, respectively) follows exactly the standards defined for the single-vehicle-product DSIRP. Results are summarised in the Tables 4.4 and 4.5.

Table 4.4: Summary of computational results for 20 products

Instances	Average cost				Average CPU (s)			
	DSIRP	DSIRP-T	DSIRPS	DSIRP-TS	DSIRP	DSIRP-T	DSIRPS	DSIRP-TS
Small	112,479.0	90,647.5	88,195.7	73,306.5	169.2	182.3	176.9	141.5
Medium	332,970.5	265,625.6	260,424.7	214,757.3	279.2	280.2	291.0	218.6
Large	601,088.3	490,336.0	472,007.6	389,921.1	307.4	247.4	293.0	263.9

We can see from Table 4.4 that the comparison results confirm that any reduction in total cost is made possible by either considering transshipment between customers or substitution of products. Moreover, considering transshipment combined with substitution enhances the performance of the overall supply chain considerably. By substituting products, less inventory is being held and by sharing further their inventory, customers are allowed to meet better their demands and decrease the lost sales and inventory costs. This is reconfirmed in the case of 40 products as it is shown in Table 4.5. On average, both DSIRP-T and DSIRPS may lead to the same results as they both can be used to

mitigate the shortage, and lower inventory and transportation costs. Combining these two emergency measures allows for a considerable reduction of costs for all the instances under consideration. Finally, the algorithm again proves efficient and competitive as it allows finding a solution in a reasonably short amount of time.

Table 4.5: Summary of computational results for 40 products

Instances	Average cost				Average CPU			
	DSIRP	DSIRP-T	DSIRPS	DSIRP-TS	DSIRP	DSIRP-T	DSIRPS	DSIRP-TS
Small	958,123.8	814,014.9	808,906.6	643,880.2	396.6	378.1	385.2	386.5
Medium	2,828,370.3	2,427,250.4	2,401,466.8	1,922,767.0	565.4	564.9	517.4	594.4
Large	5,157,985.9	4,389,631.7	4,382,510.6	3,493,632.6	1,127.3	1,176.2	1,123.3	1,192.0

4.6.5 . Computational results for other demand patterns

Intermittency of demands of products such as spare parts can be characterised by the infrequent demands that occur at irregular intervals, often of variable size. Modelling demand from constituent components, i.e. the demand size and inter-demand interval, is thus preferable. Compound theoretical distributions (which explicitly consider the combination of size and interval) are therefore commonly used in these application contexts (Conceição et al., 2015; Syntetos et al., 2012; Turrini and Meissner, 2019). In this paper, d_{pit} represents demand that each customer i has to satisfy per product p and per period t . Different distributions have been studied. We have chosen discrete distributions as they provide a better fit for intermittent demands than continuous ones. According to Syntetos et al. (2012) these distributions are: (1) Poisson Distribution (PD) for demand occurrence, combined with demands of constant size over the planning horizon. (2) Stuttering Poisson distribution (SPD) is a combination of a Poisson distribution for demand occurrence and a Geometric distribution for demand size over the planning horizon. (3) Negative Binomial Distribution (NBD) is a combination of a Poisson distribution for demand occurrence and a Logarithmic distribution for demand size over the planning horizon.

For $\zeta = 0, 1, 2, \dots$ the distribution functions can be written as:
Poisson distribution occurrence PD_λ :

$$PD_\lambda(\zeta) = \frac{\lambda^\zeta e^{-\lambda}}{\zeta!} \quad (4.24)$$

Stuttering Poisson distribution $SPD_{(\lambda, \omega)}(\zeta)$:

$$SPD_{(\lambda, \omega)}(\zeta) = \sum_{1 \leq i \leq \zeta} e^{-\lambda} \frac{\lambda^i}{i!} \binom{\zeta - 1}{i - 1} \omega^i (1 - \omega)^{\zeta - i} \quad (4.25)$$

where λ and ω are the Poisson and geometric distribution parameters.

Negative Binomial distribution $NBD_{(r, l)}(\zeta)$:

$$NBD_{(r, l)}(\zeta) = \binom{\zeta + r - 1}{\zeta} l^r (1 - l)^\zeta \quad (4.26)$$

where r is the number of successes, and l is the probability of success.

We used the Inverse Transform Sampling (algorithm 3) to generate independent and identical distributed (i.i.d.) random sample for d_{pit} realisations for each distribution under consideration.

Algorithm 3 Inverse Transform Sampling

1: **procedure** ITS(F) ▷ F is a distribution function
2: $\chi \leftarrow$ Generate random number from the standard uniform distribution in $[0, 1]$;
3: $\zeta \leftarrow F^{-1}(\chi)$
4: **end procedure**

We conduct extra experiments on 150 instances generated based on the experimental design to get more insight into the advantage of both transshipment and substitution. We consider a set of customers varying between 5 and 200, periods between 5 and 20 and a number of products of 40. For each customer, period and product, Poisson and geometric distribution parameters λ and ω , as well as the NBD parameters r and l , are random numbers generated between 0 and 1. Table 4.6 reports the summary of computational results for the different demand distributions under consideration.

Table 4.6: Computational results for the different distribution patterns

Distribution	Instances	Average cost				Average CPU			
		DSIRP	DSIRP-T	DSIRPS	DSIRP-TS	DSIRP	DSIRP-T	DSIRPS	DSIRP-TS
PD	Small	579,620.0	510,199.0	512,898.8	395,199.3	172.5	206.2	185.5	191.6
	Medium	1,717,250.6	1,514,218.5	1,515,691.4	1,157,961.0	913.9	895.7	879.7	975.9
	Large	3,094,277.0	2,717,394.1	2,692,526.4	2,104,930.6	1,785.5	1,920.2	1,671.7	1,723.1
	Average	1,797,049.2	1,580,603.9	1,573,705.6	1,219,363.6	957.3	1,007.4	912.3	963.5
SPD	Small	659,197.4	582,248.4	580,904.5	450,600.6	203.2	183.5	187.6	206.8
	Medium	1,870,371.9	1,661,277.5	1,649,561.1	1,281,203.8	201.5	747.4	947.4	939.4
	Large	3,590,243.8	3,175,741.8	3,141,875.6	2,407,310.5	1,818.8	1,996.7	1,714.0	1,744.5
	Average	2,039,937.7	1,806,422.6	1,790,780.4	1,379,705.0	741.2	975.9	949.7	963.5
NBD	Small	706,205.3	626,912.6	617,771.5	472,025.9	180.1	180.0	219.1	181.3
	Medium	2,081,957.6	1,845,195.7	1,820,708.2	1,391,390.4	739.7	827.9	982.4	690.8
	Large	3,772,317.6	3,338,938.7	3,287,841.0	2,539,575.3	1,860.2	1,800.6	1,952.2	1,690.9
	Average	2,186,826.8	1,937,015.6	1,908,773.6	1,467,663.9	926.7	936.2	1,051.2	854.3

We notice that allowing transshipment and substitution considerably reduces the total cost. When these two options are not taken into account, the supply chain experiences a high cost of inventory and loss of sales. Transshipment and substitution reduce lost sales and inventory holding costs at the level of each customer. When transshipment is permitted, results show that it offers a number of advantages: customers receiving the quantity latterly transshipped can satisfy even more demand and consequently reduce lost sales. Customers from which the transshipment is carried out, are, in counterpart, able to lower their inventory holding costs. When the substitution is also allowed along with transshipment, compared to the other configurations, we observe a considerable reduction of the costs (to about 32%). In addition to what can be received through transshipment, each customer can use the quantities of products, if compatible, that could constitute idle stock (which leads to high holding cost) to meet the demand of other products. Furthermore, by means of substitution, the quantity of products that could be transshipped can be also used as a substitute for different products. As for demand patterns, very low size variability and value of demands (as in the case of PD), regardless of the average inter-demand interval may be less stressful than the case when demands experience high variability and size. Indeed, when demands to satisfy are higher than the available quantity to promise, there would not be enough quantity for substitution and transshipment to lessen any possible lost sales. For this reason, the impact of transshipment and substitution depends on whether the distribution of demands to meet may or not be considered a stressful scenario.

4.7 . Conclusions & Perspectives

In this paper, we consider a two-level supply chain. At the upper level, a manufacturer-owned central warehouse distributes products to a given number of customers (the lower level). We model the problem as a dynamic and stochastic inventory-routing problem that considers the two flexible instruments of transshipment and substitution to mitigate shortages. We assume that lost sales are allowed when a shortage occurs. We solve the problem using a new metaheuristic which combines the mathematical modelling, the strong global search ability of the Genetic Algorithm and the self-adaptability of the Deep Q-learning. The metaheuristic is first applied to a set of 150 known instances and is found to be competitive and efficient as it enhances the best-known solutions of the single-vehicle-product DSIRP. We later solve the problem for multi-product-vehicle DSIRP. Four demand distributions have been studied: Normal distribution, Poisson distribution for demand occurrence, combined with demands of constant size, Stuttering Poisson distribution and Negative Binomial distribution. Regarding the managerial insights, for all the demand patterns under consideration, we demonstrate the benefits of promoting inventory sharing or substitutions as emergency measures to sidestep shortages. In addition, we show that combining these two flexible instruments can be a viable solution for supply chain managers aiming to improve the system's broad service level under dynamic and stochastic demands. Moreover, results show that the impact of transshipment and substitution on the overall performance depends on the size variability of demands, regardless of the average inter-demand interval.

This paper can be expanded to investigate the performance of the resolution approach in centralised and decentralised settings, either deterministic or stochastic. One can also consider non-parametric methods for demand, whereby the empirical demand distribution is instead directly constructed from the data. The effect of the forecasting method and the resulting error can also be integrated into the model. Also, a stochastic optimisation on demands such as average sample approximation could be applied (as investigated in Chapter 3). This paper assumes that the lead times are long between the facility and the central warehouse. Additionally, stochastic lead time and production rate can also be investigated. Furthermore, it would be interesting to consider a substitution rate instead of a compatibility matrix. This would allow customers to more explicitly express their preference with regard to the available products.

Acknowledgements

We would like to show our gratitude to the (Prof. Leandro C. Coelho, Ph.D., Canada Research Chair in Integrated Logistics, Laval University) for sharing valuable information about the single-vehicle-product DSIRP data-set. His help is sincerely appreciated. The authors thank the editor-in-chief, the associate editor and three anonymous referees for their constructive comments and encouragements that have helped improve our paper greatly.

5 - Modeling and solving a decentralised and deterministic multi-product multi-vehicle VRPT

Preliminary version of the paper published on Transportation Research Procedia as part of the Special Issue of the 24th Euro Working Group on Transportation Meeting, EWGT 2021, Aveiro, Portugal, <https://doi.org/10.1016/j.trpro.2022.02.064>

Achamrah, F.E., Riane, F. and Aghezzaf, E.H. (2022). Bi-level programming for modeling inventory sharing in decentralised supply chains. Transportation Research Procedia.

Full paper to be submitted to Journal of Intelligent Transportation Systems – Taylor & Francis, ISSN 15472450, Impact Factor: 4.2, Quartile: Q1

Bi-level programming for modeling inventory sharing in decentralised supply chains

Fatima Ezzahra Achamrah ^{a,b}, Fouad Riane ^{a,b}, El-Houssaine Aghezzaf ^{c, d}

^aComplex Systems and Interactions, Ecole Centrale of Casablanca; ^bLaboratoire Genie Industriel, CentraleSupélec, Paris Saclay University; ^c Ghent University, Faculty of Engineering and Architecture, Department of Industrial Systems Engineering and Product Design, ^d Industrial Systems Engineering (ISyE), Flanders Make

Abstract: This paper deals with inventory sharing and routing in decentralised supply chains. The supply chain considered in this paper consists of a single company distributing its products through a network of independent Points of Sale (POS). The problem is modelled as a 1-leader - n -followers Stackelberg game. A new mixed-integer bi-level program is developed, in which the manufacturer's central warehouse decides first on inventory levels and the distribution routes, considering each follower's (POS) response function that minimises the follower's own cost. A trade-off solution to manage conflict of interests between the parties involved in the supply chain is also proposed. To solve the mixed-integer bi-level program, an original hybrid Genetic Algorithm coupled with deep reinforcement learning is developed and used to solve a set of large-size instances. The gap analysis shows that the proposed hybrid algorithm performs relatively well and that inventory sharing allows the network to improve its service level.

5.1 . Introduction

To increase service levels and reduce logistics costs, manufacturers tend to encourage, through lateral transshipment (LT) policy, sharing of inventories between a network of customers (e.g., points of sale (POS), retailers) belonging to the same echelon (Chiou, 2008; Peres et al., 2017; Tarhini et al., 2019; Wang et al., 2019). To successfully achieve such economies of scale, each LT policy must balance conflicting interests and manage inventories and deliveries. Moreover, it has to choose the right LT price so that both manufacturer and customers would benefit from LT and its related cost would not exceed the profit to be made (Shao et al., 2011; Hezarkhani and Kubiak, 2010; Atan et al., 2018). This is the case of decentralised supply chains in which the same manufacturer does not own customers' locations, and each decision-maker or level of the supply chain works to optimise its costs (Liao et al., 2020; Li et al., 2020).

In the literature on vehicle routing (VRP) or IRP with LT, the problem is thoroughly studied in its centralised version (Coelho et al., 2012b; Achamrah et al., 2021; Lefever et al., 2018; Timajchi et al., 2019; Mirzapour Al-e-hashem and Rekik, 2014). Moreover, to simplify the optimisation problems, most papers assume that LT operations are outsourced and performed by another carrier; only consider decision variables that determine nodes to visit and periods when LT may occur (Achamrah et al., 2022b). However, LT often occurs with independent customers' locations not owned by the manufacturer. On decentralised decision making, LT has been extensively studied in the context of news vendor-related supply chains (Rudi et al., 2001; Yan and Zhao, 2011; Arikani and Silbermayr, 2018; Zhao et al., 2016). According to Hanany et al. (2010), there exist two variants to be distinguished: vertical and horizontal systems. Vertical if the manufacturer supplies to a

single downstream company with a set of stores (a chain store). Using a deterministic demand pattern, [Dong and Rudi \(2004\)](#) demonstrates that the manufacturer benefits from LT under reasonable assumptions. [Zhao et al. \(2005\)](#) generalises the results of [Dong and Rudi \(2004\)](#) to include stochastic demands. A horizontal decentralised supply chain with LT concerns customers' locations not owned by the same company. Two approaches are being used. The first one adopts a non-cooperative game framework. [Rudi et al. \(2001\)](#) and [Hu et al. \(2007\)](#) compare equilibrium inventory levels with and without LT. [Rong et al. \(2010\)](#) consider LT in multiple periods. [Jiang and Anupindi \(2010\)](#) compare the LT game with another game in which consumers are allowed to choose among retailers. The upstream (manufacturer) is not considered in the study in all these papers. [Grahovac and Chakravarty \(2001\)](#) study the case of an inventory control policy for a supply chain in which a manufacturer acts as a Stackelberg leader. The other approach uses a cooperative game framework ([Anupindi et al., 2001](#); [Granot and Sošić, 2003](#); [Sošić, 2006](#)). Particularly, these papers address the "coopetition" between customers' locations (e.g., POS, retailers). That is, customers define their inventories unilaterally before satisfying the demands of their customers and then collectively decide how to share their inventory. [Huang and Sošić \(2010\)](#) compare the two approaches and conclude that, based on the model's parameters, each approach can be more effective than the other in gaining additional benefit from transshipment. However, the LT has been usually treated as the single simplified manufacturer multiple customers inventory management problem in which vehicle routing, delivery scheduling decisions, and the relative power of all parties in the decision-making are not considered.

This paper makes several contributions to the literature. First, it studies a multi-product, multi-vehicle Vehicle Routing Problem (VRP) with LT and Inventory Management (VRP-TIM) in a decentralised supply chain. The supply chain consists of a manufacturer's central warehouse (CW) that distributes products through a network of independent customers. Demand for finite horizon planning is deterministic but time-varying. Customer product delivery is carried out using a homogeneous capacitated fleet of vehicles. Moreover, unlike other research, our approach considers that LT is not outsourced. That is, its related decision is integrated into the design of vehicle routing. Each player's preliminary decision is to enhance the service level while maintaining a minimum cost, including transportation, inventory, lost sales, and LT. Second, the paper suggests a trade-off solution to manage conflict of interests between the supply chain's players. Indeed, as a part of the collaboration, the CW and the customers may agree to incur each their own holding cost as well as a part of the cost of lost sales associated with the products shortage and a part of the cost of LT. The CW incurs, in turn, the vehicle routing cost for regular shipments. Therefore, inventory sharing can only attract all players if the LT and lost sales costs share are optimally defined. Such policies are often optimised locally as both players aim at reducing their objective functions that are narrowly defined. Therefore, solving such problems requires hierarchical decision-making, which belongs to the multilevel optimisation family ([Grahovac and Chakravarty, 2001](#)). Moreover, many automotive, high-tech and fashion enterprises can benefit from this research. Online retailing companies with several independent stock-keeping locations may particularly benefit, as customers usually have no preference for the location from which their products are delivered as long as these latter are made available at the right time and place. Finally, to handle the combinatorial complexity of the model, an original hybrid Genetic Algorithm (HGA) coupled with deep reinforcement learning is proposed.

The remaining of this work is structured as follows. In Section 2, related works are presented. Problem description and formulation are provided in Section 3. Section 4 describes the resolution approach, while Section 5 shows the computational experiments. Finally, conclusions and perspectives are presented in Section 6.

5.2 . Related work

Two streams that are of great importance have been examined. The first investigates the pickup and delivery problem (PDP) in which pickup and delivery operations are optimised along with inventory management or/and LT. The second examines bi-level optimisation as a modelling approach for decentralised decision-making and presents the resolution methods.

5.2.1 . Pickup and Delivery Problem

This section presents the class of problems that combines vehicle routing and inventory management decisions and includes problems where products are picked up from different origins and transported to other destinations, i.e., problems addressing the optimisation of pickup and delivery operations and inventory management. The reader, if interested, is referred to [Berbeglia et al. \(2007\)](#) and [Battarra et al. \(2014\)](#) for a thorough description of the other variant of PDP.

Regarding literature on VRP with Pickup and Delivery (PD), two classes of customers are to be distinguished: linehaul and backhaul ([Wade and Salhi, 2002](#)). Linehaul if products are requested to be delivered from the depot, while backhaul if products are picked up from and returned to the depot. These two classes are further divided into sub-classes concerning the way these customers are served:

1. VRP with backhauls when all linehauls are visited before backhauls ([Wassan et al., 2017](#)).
2. VRP with mixed PD if linehauls and backhauls can be visited in any order ([Avci and Topaloglu, 2015](#)).
3. VRP with simultaneous PD if a customer can be both linehaul and backhaul ([Gong et al., 2018](#)).
4. VRP with split PD when deliveries and pickups are split so that the customers can be visited twice either by the same vehicle or by another ([Polat, 2017](#)).

Another research stream which is relevant to this paper is the one that studies the Inventory routing problem (IRP) with PD (IRPPD) such that quantities of pickup and delivery are decided based on an inventory policy, and the decision-making is centralised based on the Vendor Managed Inventory (VMI). The most relevant papers which study IRPPD are related to maritime applications ([Christiansen and Fagerholt, 2014](#); [Christiansen et al., 2013](#)). On-road transportation, a recent paper of [Archetti et al. \(2018\)](#) studied a single-product and single-vehicle problem IRPPD in which a commodity must be picked up from pickup customers and delivered to delivery customers over a given planning horizon with a single capacitated vehicle. [Archetti et al. \(2020\)](#) extend this work to the multiple vehicle case. In both papers, it is assumed that the role of each customer does not change. That is, the role of pickup or delivery customers remains unchanged throughout the time horizon. [Iassinovskaia et al. \(2017\)](#) study IRP with simultaneous pickups and deliveries of the returnable transport items in a two-level supply chain. The most closed paper to us is the paper of [van Anholt et al. \(2016\)](#). Unlike other articles, the authors consider two PD structures in their model: one to many to 1 structure, which accounts for a product movement from the depot to automated teller machines to the depot, and the M-M structure, which refers to commodity transfers among the machines and which is assumed to be visited either for a pickup or a delivery operation. However, in the IRPPD under consideration, routes are mainly constructed for pickup and delivery; while this deals with a decentralised supply chain, LT is considered as a complementary measure

to sidestep shortage, and there are multiple products. Moreover, the role of customers can change over time horizon. That is, in a given period, each customer can be a pickup node, delivery node, or both, depending on its inventory level.

5.2.2 . Bi-level optimisation

Game Theory can be described as "the study of mathematical models of conflict and cooperation between intelligent, rational decision-makers" (Myerson, 1991). The outcome of a competitive game is generally an equilibrium (e.g., Cournot, Nash, Stackelberg) where none of the decision-makers would benefit by modifying their decision. In this work, we focus on some non-cooperative games called Stackelberg Games, sequential non-zero-sum games involving two players. Connected to game theory, bi-level problems can be seen as the mathematical programming counterpart of Stackelberg games introduced by H. von Stackelberg in 1952. These games can be mathematically modelled as two nested optimisation problems, also referred to as bi-level problems (Sinha et al., 2017). In the bi-level programming model, the first level decision-maker (Upper Level: UL) is called the leader and the second level decision-maker, the follower (Lower Level: LL) (Colson et al., 2007). Each decision-maker tries to optimise his objective function regardless of the purpose of the other. Nevertheless, each decision-maker's choice influences the value of the objective function and the other level's decision-making space (Dempe and Zemkoho, 2020). Therefore, the UL must choose an answer as the optimal solution in bi-level programming, which is also optimal for the decision-makers on the second level. This nested structure implies that a feasible solution at the UL should be optimal for the LL problem (Colson et al., 2007). This is the reason why bi-level optimisation problems are challenging to solve. Many practical problems involving a bi-level structure have been studied in the literature. For instance: inventory management with LT in a decentralised supply chain (Grahovac and Chakravarty, 2001) and VRP (Marinakos et al., 2007).

On the other hand, Ben-Ayed (1988) stress that as the Bi-Level Programming Problem (BLPP) is non-convex, it is difficult to solve since the simplest case is an NP-hard problem. The methods developed to solve BLPP can be classified into two main categories: classical and evolutionary methods (Feng et al., 2018). The classical or exact methods include vertex enumeration methods (Bard and Falk, 1982), Kuhn-Tucker transformation methods (Shi et al., 2005), and penalty function techniques (White and Anandalingam, 1993). Classical methods can address only differentiable and convex problems. Furthermore, the BLPP solution requirements produce disconnectedness and non-convexity even with simple problems. Zhang et al. (2016) propose a survey on multilevel optimisation problems and their solution techniques. They highlight that exact methods have been used to solve a particular case where BLPP is used and have not been used to solve various multilevel programming problems, especially for large-scale problems. Exact solutions thus show some drawbacks when it comes to large-scale problems, as they continue to rely on consistency, differentiability, and convexity.

Conceptually, the evolutionary methods differ from the classical methods. They can address differentiable and convex optimisation problems. Evolutionary methods are inspired by the human, biological, genetic, insect swarm, and neuro-biological processes Yang (2010). They cover metaheuristics such as GA, simulated annealing, particle swarm's optimisation, ant colony optimisation, neural networks, immune systems, etc. Applications can be found in Carrasqueira et al. (2017), Li et al. (2014) and Wang et al. (2008). Among these metaheuristics, particle swarm optimisation (PSO) and GA have been proven to be more efficient and convenient in the resolution of BLPP. Kuo and Han (2011) and (Li et al., 2014) highlight that the PSO algorithm is more practical in solving nonlinear and large-sized problems. Ma et al. (2016) and Jia et al. (2014) use a GA algorithm to solve a bi-level supply chain planning problem and showed its high computational perfor-

mance. In [Amirtaheri et al. \(2017\)](#), it is stressed that PSO and GA algorithms manifest high computational efficiency: GA is ultimately discrete and can be applied to discrete optimisation, especially for decision problems with 0-1 decision variables. At the same time, PSO is inherently continuous and can be used in continuous optimisation. GA typically applies a crossover operator to two solutions that play a key role in the algorithm. Often, a fixed probability mutation operator is used that changes randomly the unique content, and a selection of performed which is generally rendered probabilistically or proportionally ([Moscatto et al., 1989](#)). Generation in this algorithm is the replacement of picked solutions. That is, the parents are substituted by their children. The literature clearly shows that GA is very robust in dealing with nonconvex and nondifferential problems, owing to its simplicity and explicit parallelism. For the above reasons, a hybrid GA is used to solve our bi-level program for large-scale instances.

5.3 . Mathematical formulation

5.3.1 . Bi-level integer program

Since the supply chain is decentralised, we formulate the problem as a bi-level integer program (BLP). We assume that the manufacturer's CW acts as a Stackelberg leader (Upper Level: UL) that chooses stocking levels and routes to be built, knowing each customer's (Lower Level: LL) response function arising from its minimisation of the total cost. In addition, we define parameter $0 \leq \epsilon \leq 1$ to denote the relative share of the cost of lost sales that is incurred by the customer and $0 \leq \beta \leq 1$ to denote the relative share of the transshipment costs that is incurred by customer receiving products. The CW incurs the remaining parts $(1 - \epsilon)$ and $(1 - \beta)$. The parameters ϵ and β can thus be seen as measures of the relative power of the partners in the supply chain.

Table 5.1: Model's notation summary

V	set of nodes including CW
P	set of products
K	set of available vehicles
H	set of periods
Parameters	
C_i	maximum inventory capacity at node $i \in V$
I_{pi0}	inventory level at the beginning of the planning horizon at each node i and product p
D_{pi}	demand to be satisfied per customer $i \in V_0$ and product p
Q	capacity of vehicle
c_k	fixed transportation cost for each vehicle k
d_{ij}	distance in km between $(i, j) \in E$
h_{pi}	unit inventory holding cost for each anode $i \in V$ and product p
a_{pij}	unit transshipment cost of a product p from a node $i \in V_0$ to node $j \in V_0$
s_{pi}	loss sales cost associated to the product p at the level of a customer $i \in V_0$
f_{pt}	quantity of product p produced at the CW at period t
Decision variables	
I_{pit}	inventory level of product p at node $i \in V$ at the end of period t
Q_{pit}	quantity of product p directly shipped from the CW to node $i \in V_0$ in period t
q_{pijkt}	quantity of product p shipped from node $i \in V_0$ to node $j \in V_0$ by vehicle k in period t . It includes regular shipment from CW and transshipment between customer
w_{pijkt}	quantity of product p transshipped from node $i \in V_0$ to node $j \in V_0$ by vehicle k in order to sidestep the shortage of the product p in period t
y_{pit}	lost sales quantity of product p at node $i \in V_0$ in period t
z_{kt}	equal to 1 if the vehicle k is used in period p , 0 otherwise
$x_{ijk t}$	equal to 1 if the arc $(i, j) \in E$ is visited by a vehicle k in period t , 0 otherwise

The related BLP is defined on a graph $G = (V, E)$ where $V = \{0, \dots, n\}$ the vertex set and $E = \{(i, j) : i, j \in V, i \neq j\}$ is the edge set. Let $V_0 = V \setminus \{0\}$ be the set of n customers (LL) and 0 be the vertex representing the CW (UL). Both the LL and UL incur an inventory holding cost h_{pi} per product $p \in P = \{1, \dots, m\}$. Each has a maximum inventory holding capacity C_i . The length of the planning horizon is T with discrete periods $t \in H = \{1, \dots, T\}$. We assume that at the beginning of the planning

horizon, the current inventory-levels for each product p I_{pi0} , expressed in terms of Stock Keeping Unit (SKU), are known for each $i \in V$. D_{pit} is the demand a point of sale $i \in V_0$ has to meet for each period t and product p . Let $k \in K = \{1, \dots, u\}$ be the available set of homogeneous vehicles. Each vehicle has a capacity Q (in equivalent SKU), with a fixed transportation cost per km c_k . A distance d_{ij} (expressed in km) is associated for all $(i, j) \in E$. The unit cost associated with transshipping a product p from a customer i to a customer j is a_{pij} . s_{pi} is the lost sale cost associated to product p at the level of the customer i . f_{pt} the quantity of product p produced by the CW at period t . The model's notation summary is given in table 5.1. The BLP for VRP-TIM can be written as:

- Upper Level

$$\begin{aligned} \min \sum_{t \in H} \sum_{p \in P} h_{p0} I_{p0t} + \sum_{t \in H} \sum_{k \in K} c_k \sum_{i \in V} \sum_{j \in V, i \neq j} d_{ij} x_{ijkt} + \\ (1 - \epsilon) \sum_{t \in H} \sum_{k \in K} \sum_{i \in V_0} \sum_{j \in V_0, i \neq j} \sum_{p \in P} a_{pij} w_{pijkt} + (1 - \beta) \sum_{t \in H} \sum_{i \in V_0} \sum_{p \in P} s_{pi} y_{pit} \end{aligned} \quad (5.1)$$

Subject to:

$$I_{p0t} = I_{p0t-1} - \sum_{i \in V_0} Q_{pit} + f_{pt} \quad \forall p \in P, t \in H \quad (5.2)$$

$$\begin{aligned} Q_{pjt} - \sum_{k \in K} \sum_{i \in V_0, i \neq j} (w_{pjikt} - w_{pijkt}) = \sum_{k \in K} \sum_{i \in V_0, i \neq j} (q_{pijkt} - q_{pjikt}) \\ \forall p \in P, j \in V_0, t \in H \end{aligned} \quad (5.3)$$

$$\sum_{p \in P} q_{pi0kt} = 0 \quad \forall i \in V_0, k \in K, t \in H \quad (5.4)$$

$$\sum_{p \in P} I_{p0t} \leq C_0 \quad \forall t \in H \quad (5.5)$$

$$\sum_{p \in P} q_{pijkt} \leq Q z_{kt} \quad \forall (i, j) \in E, k \in K, t \in H \quad (5.6)$$

$$\sum_{i \in V} x_{ijkt} = \sum_{i \in V} x_{jikt} \quad \forall j \in V_0, k \in K, t \in H \quad (5.7)$$

$$\sum_{i \in V} \sum_{k \in K} x_{ijkt} \leq 1 \quad \forall j \in V_0, k \in K, t \in H \quad (5.8)$$

$$\sum_{j \in V_0} x_{0jkt} = z_{kt} \quad \forall k \in K, t \in H \quad (5.9)$$

$$\sum_{k \in K} z_{kt} \leq u \quad \forall t \in H \quad (5.10)$$

- Lower Level

$$\begin{aligned} \min \sum_{t \in H} \sum_{i \in V_0} \sum_{p \in P} h_{pi} I_{pit} + \epsilon \sum_{t \in H} \sum_{k \in K} \sum_{i \in V_0} \sum_{j \in V_0, i \neq j} \sum_{p \in P} a_{pij} w_{pijkt} + \\ \beta \sum_{t \in H} \sum_{i \in V_0} \sum_{p \in P} s_{pi} y_{pit} \end{aligned} \quad (5.11)$$

Subject to:

$$I_{pit} = I_{pit-1} + Q_{pit} - D_{pit} - \sum_{k \in K} \sum_{j \in V_0, i \neq j} (w_{pjikt} - w_{pijkt}) + y_{pit}$$

$$\forall i \in V_0, p \in P, t \in H \quad (5.12)$$

$$\sum_{p \in P} I_{pit} \leq C_i \quad \forall i \in V_0, t \in H \quad (5.13)$$

$$\sum_{k \in K} \sum_{j \in V_0} w_{pijkt} \leq I_{pit-1} \quad \forall (i, j) \in E, k \in K, t > 1 \quad (5.14)$$

The objective function (5.1) at the UL minimises its total cost. The first sum corresponds to the inventory cost. The second sum is for transportation costs. The third is the total shared cost of transshipment $(1 - \epsilon)$. The last sum is the shared cost of lost sales $(1 - \beta)$. Constraints (5.2) indicate the conservation conditions of inventory at the CW over successive periods. Constraints (5.3) express the conservation of flows (inflows and outflows) at each customer j . Constraints (5.4) guarantee that vehicles are emptied when returned to the CW at the end of a period. Constraints (5.5) guarantee that inventory levels at the CW do not exceed the maximal holding capacity. Constraints (5.6) state that the quantities transported do not exceed the vehicle capacity. Constraints (5.7) stipulate that when a vehicle k visits the customer j in period t , the customer j must be left in period t . Constraints (5.8) ensure that the most a customer is visited once by the vehicle k per period. Constraints (5.9) stipulate that only vehicles shipping products are used. Constraints (5.10) indicate that only available vehicles are used. The objective function (5.11) at the LL minimises the customer's costs. The first sum corresponds to the inventory cost. The second is the total shared cost of transshipment (ϵ) . The last sum is the total shared cost of lost sales (β) . Constraints (5.12) indicate the conservation conditions of inventory at the customer over successive periods. The constraints (5.13) guarantee that inventory levels at each customer do not exceed the maximal holding capacity. Constraints (5.14) state that the quantity latterly transshipped from customer i at a period t does not exceed the inventory level at the beginning of the period.

5.3.2 . Reformulation of LL model using Karush-Kuhn-Tucker conditions

The traditional idea of the reformulation of the BLP is to substitute the LL mathematical program by its Karush-Kuhn-Tucker (KKT) conditions as sufficient and necessary optimality conditions (Bouza Allende and Still, 2012). Let σ_{pit}^1 , σ_{it}^2 and σ_{pit}^3 be the dual variables associated with LL's constraints. We use the Lagrangian function to compute the relative complementary conditions and dual feasibility. The BLP is converted into a single linear program subject to the UL feasibility constraints, LL primal and dual feasibility constraints, and complementary conditions. The BLP for VRP-TIM is therefore converted into a single linear program in the following format:

$$\begin{aligned} \min \quad & \sum_{t \in H} \sum_{i \in V} \sum_{p \in P} h_{pi} I_{pit} + \sum_{t \in H} \sum_{k \in K} c_k \sum_{i \in V} \sum_{j \in V, i \neq j} d_{ij} x_{ijkt} + \\ & \sum_{t \in H} \sum_{k \in K} \sum_{i \in V_0} \sum_{j \in V_0, i \neq j} \sum_{p \in P} a_{pij} w_{pijkt} + \sum_{t \in H} \sum_{i \in V_0} \sum_{p \in P} s_{pi} y_{pit} + \\ & \sum_{t \in H} \sum_{i \in V_0} \sum_{p \in P} \sigma_{pit}^1 D_{pit} + \sum_{t \in H} \sum_{i \in V_0} \sigma_{it}^3 C_i \end{aligned} \quad (5.15)$$

Subject to:

Upper level feasibility: constraints (5.2)– (5.10)

Lower level primal feasibility: constraints (5.12)– (5.14)

Substituted complementary conditions:

$$\begin{aligned} & \sum_{i \in V_0} \sum_{t \in H} \sum_{p \in P} h_{pi} I_{pit} + \epsilon \sum_{t \in H} \sum_{k \in K} \sum_{i \in V_0} \sum_{j \in V_0, i \neq j} \sum_{p \in P} a_{pij} w_{pijkt} + \beta \sum_{t \in H} \sum_{i \in V_0} \sum_{p \in P} s_{pi} y_{pit} \\ & = - \left(\sum_{i \in V_0} \sum_{p \in P} \sum_{t \in H} \sigma_{pit}^1 D_{pit} + \sum_{t \in H} \sum_{i \in V_0} \sigma_{it}^3 C_i \right) \end{aligned} \quad (5.16)$$

Lower level dual feasibility:

$$\sigma_{it}^2 - \sigma_{pit}^3 \leq h_{pi} \quad \forall i \in V_0, p \in P, t \in H \quad (5.17)$$

$$-2\sigma_{pit}^1 + \sigma_{pit}^3 \leq \epsilon a_{pij} \quad \forall i \in V_0, p \in P, t \in H \quad (5.18)$$

$$\sigma_{pit}^1 \leq \beta s_{pi} \quad \forall i \in V_0, p \in P, t \in H \quad (5.19)$$

5.4 . Resolution approach

This section describes the resolution approach adopted to solve the problem at hand.

5.4.1 . Hybrid Genetic Algorithm coupled with reinforcement learning

The bi-level problems are intrinsically hard, even for convex levels. The simplest bi-level linear programs have been proven strongly NP-Hard (Zhang et al., 2016). The complexity induced by multiple levels and/or multiple objectives makes exact approaches non-efficient in tackling large-sized problems. For this reason, researchers turned towards metaheuristics. Among the metaheuristic algorithms, GA has proved to be the most practical and quite robust in dealing with discrete problems (Amirtaheri et al., 2017; Jia et al., 2014). On the other hand, GA is fast at global search but slow to converge (Nia and Alipouri, 2009). Furthermore, local search heuristics such as Variable Neighborhood Search (VNS) are good at fine-tuning but often fall into local optimum. In this paper, a hybrid metaheuristic which combines the properties of GA and VNS is used. Indeed, GA performs a global search to escape from the local optimum, whereas VNS is used to conduct fine-tuning. Finally, as in Achamrah et al. (2021), deep reinforcement learning is used further to enhance the speed and convergence of the algorithm and to cope with GA's limitation in terms of speed of convergence highlighted in Achamrah et al. (2022b,d).

5.4.1.1 . General description of HGA

As shown in the Algorithm 4, the first step consists of applying GA for UL. First, we generate randomly an initial population of chromosomes representing the routing decisions of the UL (a sequence of customers to visit). Once generated, the related mixed-integer program is solved. For each chromosome, a fitness value is computed concerning the objective function of the UP. Next, a double-point crossover is performed. A VNS procedure is used to replace any child created with its better neighbour. A reverse mutation is performed depending on the corresponding mutation rate. VNS is again applied to replace a mutated child with its better neighbour. The next population from the population size of the best available solutions is chosen. The best available solution based on the computed fitness is saved. All the steps above are repeated until the maximum number of iterations is reached. Based on the best solution for the upper level, the above stages are performed to find the best solution for the lower level. Based on the best solution

of the LL, an initial solution for the upper level is constructed. All the steps mentioned above are repeated until no solution improvement is noted.

Algorithm 4 : HGA

- 1: Generate an initial population.
 - 2: Calculate the fitness value of each chromosome according to UL's objective function.
 - 3: Execute the cross-over operator.
 - 4: Use VNS to replace any child the cross-over generates with its best neighbour.
 - 5: Conduct mutation process.
 - 6: Perform a mutation for the insertion.
 - 7: Apply the VNS method to substitute any child with its best neighbour generated by mutation.
 - 8: Measure the fitness value of all strengthened chromosomes within the current population; select the next population with the best possible solutions from the population size and save the best relative solution.
 - 9: Repeat steps 3 to 7 until the maximum number of iterations is reached.
 - 10: Depending on the best UL solution, repeat the above steps for the lower level.
 - 11: Depending on the best LL solution, build an initial solution for UL.
 - 12: Repeat steps 2 to 11 until no improvement of the solution is noted or a time limit is reached.
-

Now we describe these steps in detail.

5.4.1.2 . Chromosomes encoding

A chromosome is represented as a bit string in which all the possible information regarding vehicles' routing, inventory levels and quantities to be transshipped are presented. In this paper, a chromosome of UL is a one-dimensional array of integer values representing the nodes (customers) to be visited. Each chromosome (χ) is partitioned into several sub-sequences, each representing a constructed route assigned to a given vehicle and in a given period. As for LL, we choose the representation space of the relative chromosome ξ to be expressed as a Cartesian product of allele sets A^l , with $l = \{1, 2\}$:

$$\xi = A^1 \times A^2 \tag{5.20}$$

The algorithm uses a 8-bit scheme $A^l = \{a_1^l a_2^l a_3^l a_4^l a_5^l a_6^l a_7^l a_8^l; a_i^l \in \{0, 1\} / l = \overline{1, 8}\}$, so that transshipped quantity from a customer takes an integer value less than or equal to its inventory level at the beginning of the period minus the demands it has to satisfy. Next, each chromosome is re-converted to its corresponding variables x_{ijkt} , I_{pit} and w_{pijkt} to be able to calculate the fitness value (F) according to each level's objective function (OF). This fitness is expressed as:

$$F = \frac{1}{OF} \tag{5.21}$$

5.4.1.3 . Genetic operators

In this algorithm, as in [Achamrah et al. \(2021, 2022b\)](#), the following operators are used:

- Cloning operator consists of selecting a member of each generated population according to its fitness and making a copy. This guarantee that the best members of the current population will be present in the next. Finding the trade-off between the algorithm's performance and speed led us to choose the best 30% of chromosomes in the current population to be copied to the next generation.
- Parent selection operator, which uses a binary tournament selection process that starts with creating two pairs of chromosomes. Every pair consists of two chromosomes randomly selected from the current population. The two best chromosomes for each pair are chosen for crossover operations. This process results in two children, each counted in the new population.
- Crossover operator is essential to mate the pairs of chromosomes to produce their offspring. In this paper, the two-point crossover is chosen, and the contents between these points are exchanged between two mated parents. A crossover is performed based on a probability P_C .
- Mutation operator is the second operator used for exploring new neighbours. The idea is to produce random bit alterations in different chromosomes. A mutation process is performed using a P_M probability.

5.4.1.4 . VNS procedure

VNS is a metaheuristic for solving combinatorial and global optimisation problems. Its basic idea is a systematic change of neighbourhood within a descent phase to find a local optimum and a perturbation phase to get out of the corresponding valley ([Hansen et al., 2019](#)). The algorithm proceeds as described in Algorithm 5. With the function *Shake* represented in Line 4, a neighborhood x' is randomly generated from the k th neighborhood of x , i.e., $x' \in N_k(x)$. Its steps are given in Algorithm 6, where it is assumed that points from $N_k(x)$ are $\{x^1, \dots, x^{|N_k(x)|}\}$. Next, a Variable Neighborhood Descent (VND) procedure is applied if a neighbourhood is changed. Its steps are presented in the Algorithm in 7. It is worth noting that the final solution should be a local minimum regarding all k_{max} neighbourhoods; thus, the chances of reaching a global one are larger than using a single structure.

Algorithm 5 : VNS

```
 $t \leftarrow 0$   $t < t_{max}$   $k \leftarrow 1$   $k < k_{max}$   $x' \leftarrow Shake(x, k)$   $x'' \leftarrow$   
 $VND(x, k)$   $OF(x'') < OF(x)$   $x \leftarrow x''$ ;  $k \leftarrow 1$  // make a move  
 $k \leftarrow k + 1$  //next neighborhood
```

Algorithm 6 : Shake procedure

```
 $\omega \leftarrow [1 + Rand(0, 1) \times |N_k(x)|]$   $x' \leftarrow x^\omega$ 
```

Algorithm 7 : VND

$k \leftarrow 1$ $k < k_{max}$ $x' \leftarrow \underset{y \in N_k(x)}{\operatorname{argmin}} OF(y)$ // find the best neighbour in $N_k(x)$ $OF(x'') < OF(x)$ $x \leftarrow x''$; $k \leftarrow 1$ // make a move $k \leftarrow k + 1$ //next neighborhood

5.4.1.5 . Constraints violation

A basic penalty strategy is adopted to respect the model's constraints in the HGA. That is, during the generation of the random initial solutions and through the crossover and mutation schemes, the feasibility of each chromosome is checked considering the constraints of the model. If an infeasibility arises in the solution, then a penalty is associated with the value of the fitness functions of the corresponding chromosome [Achamrah et al. \(2021\)](#). In this way, infeasible chromosomes have a lower probability of being part of the next generation of chromosomes.

5.5 . Computational experiments

5.5.1 . Data set and parameters tuning

This section summarises the computational experiments performed on a set of instances randomly generated by [Coelho and Laporte \(2013a\)](#) for a multi-products IRP. For each instance, the number of customers varies between 10 and 50, the number of both products and vehicles varies between 1 and 5, and finally, the number of periods varies between 3 and 5. Product availability at the CW is a multiple of a number randomly generated according to a discrete uniform distribution in the interval $[50, 140]$, and the maximum inventory level is a multiple of a number drawn randomly from $[150, 200]$. The initial inventory level is a randomly generated number in the interval $[100, 150]$. Holding costs are randomly generated from a continuous uniform distribution in the interval $[0.02, 0.2]$. The reader is referred to their paper for further details, and the dataset can be downloaded from <http://www.leandro-coelho.com/instances/>.

As in [Achamrah et al. \(2021, 2022b\)](#), to fine-tune the parameters, we use the Irace package (see Table 5.2). All optimisation steps are carried out on a personal computer (MacBook Pro, macOS Cataline, CPU 3.3 GHz Quad-Core Intel Core i7, 8 GB of RAM). The problem is solved using the branch-and-cut solver of CPLEX 12.9 (academic version), Python 3.7 and Pytorch. For each instance, ten independent runs are performed using HGA, and the average of the results is recorded. We refer to the instances using the following notation: [number of products] P [number of customers] N [number of periods] T, e.g., 2P5N5T refers to the instance where two products are shipped to a set of 5 customers over a planning horizon of 5 days.

Table 5.2: HGA's parameters

Parameter	Range	Chosen value
Population size (UL / LL)	[100,140]	100
Maximum number of iterations	[100,140]	100
Mutation probability	[0.10,0.18]	0.15
Crossover probability	[0.5,0.55]	0.52
Cloning probability	[0.2,0.4]	0.3

Table 5.3: DQ's parameters

Parameter	Value
Exploration rate	[0,1]
Discount rate	0.33
C	100

5.5.2 . Primary insights into ϵ and β optimal value

Considering a Stackelberg game, the parameters ϵ and β can be seen as measures of the relative power of the partners in the supply chain under consideration. The policies at the level of the CW and the customers are locally optimal since all the partners minimise their narrowly defined objectives. The objective is to determine the scenario in which transshipment is beneficial, and the total cost is minimised for all the partners. That is, determine ϵ^* the fair shares relative to transshipment cost given the shared parameter β of lost sales cost. Without loss of generality, we consider an instance consisting of 10 customers and five products. For a given value of β , we vary $\epsilon \in \{0; 0.2; 0.4; 0.6; 0.8; 1\}$ and observe the variation of the total cost at the level of CW and customers. The intersection of the two curves makes it possible to determine ϵ^* (see Table 5.4).

From Table 5.4, we can see that the CW tends to become relatively more interested in instituting inventory sharing schemes as the customer incurs higher LT and shortage costs. Indeed, whenever the CW incurs a major part of the cost of lost sales, he becomes more cautious about sharing the cost of LT. And vice versa. If he incurs a large part of LT cost, it is in his interest to incur a small percentage of the cost of lost sales. As a result, a fair share of costs and inventory sharing is efficient if only they make at least one party strictly better-off while making no one else worse-off.

Table 5.4: Optimal shares relative to the transshipment for a given value of share of lost sales

β	0	0.2	0.4	0.6	0.8	1
ϵ^*	~ 0.95	~0.87	~0.80	~0.76	~0.68	~0.6

5.5.3 . Results on small and large instances

This section aims to provide insight into the benefits of transshipment and the representativeness of the results. For each instance, we consider two scenarios in which we compare the bi-level program with (VRP-TIM) and without transshipment (VRP-IM). Tables 5.5, 5.6, and 5.7 summarise the results of the comparison between CPLEX and HGA in terms of total costs (TC). It also provides the gap (GAP) computed regarding the total costs obtained using CPLEX (with a time limit of 24 hours). For each instance under consideration, we remade the same tests done in the previous section by varying ϵ and β to determine the optimal ϵ^* and β^* ; corresponding to the lowest cost recorded so far. Tables 5.9, 5.8 and 5.10 summarise the results for each instance under consideration. For all instances, the breakdown of costs is provided, namely: Transportation (T), Inventory (I), Lost sales (LS) and transshipment (Ts). Tables also provide the service levels computed regarding the satisfied and lost demands and report CPU time in second (CPU) needed to solve the models using HGA. From Tables 5.9, 5.8 and 5.10, results show that transshipment is overall efficient as it reduces lost sales and enhances service levels. It also shows that as a leader, the CW tends to become relatively more interested as the customer incurs higher costs relative to transshipment and lost sales. In addition, any of the involved parties can end up better off once transshipment is allowed. Hence, depending on the value of the parameters ϵ^* and β^* , inventory sharing may work both in

Table 5.5: Results of comparison between CPLEX and HGA on small and relatively large instances (T=3, P=[1;5])

Instance	Model	CPLEX (24 hours)	HGA	GAP (%)
		TC	TC	
1P10N3T	VRP-IM	18516	18552	0.2
	VRP-TIM	9187	9233	0.5
3P10N3T	VRP-IM	29513	29810	1.0
	VRP-TIM	17262	17326	0.4
5P10N3T	VRP-IM	31768	31859	0.3
	VRP-TIM	21266	21402	0.6
1P20N3T	VRP-IM	34593	34885	0.8
	VRP-TIM	21355	21530	0.8
3P20N3T	VRP-IM	42537	42890	0.8
	VRP-TIM	30651	30950	1.0
5P20N3T	VRP-IM	57323	57775	0.8
	VRP-TIM	34759	34855	0.3
1P30N3T	VRP-IM	75214	75413	0.3
	VRP-TIM	44183	44526	0.8
3P30N3T	VRP-IM	84497	84502	0.0
	VRP-TIM	61422	61765	0.6
5P30N3T	VRP-IM	146285	147026	0.5
	VRP-TIM	120758	121877	0.9
1P40N3T	VRP-IM	298157	300525	0.8
	VRP-TIM	211125	213115	0.9
3P40N3T	VRP-IM	532164	532675	0.1
	VRP-TIM	312668	314693	0.6
5P40N3T	VRP-IM	1045161	1047582	0.2
	VRP-TIM	560721	564945	0.7
Average		160045	160821	0.6

Table 5.6: Results of comparison between CPLEX and HGA on small and relatively large instances (T=5, P=[1;5])

Instance	Model	CPLEX (24 hours)	HGA	GAP (%)
		TC	TC	
1P10N5T	VRP-IM	54906	57374	4.3
	VRP-TIM	25520	26571	4.0
3P10N5T	VRP-IM	83486	86809	3.8
	VRP-TIM	41948	45754	8.3
5P10N5T	VRP-IM	95773	100671	4.9
	VRP-TIM	64473	68463	5.8
1P20N5T	VRP-IM	83858	89549	6.4
	VRP-TIM	63147	63336	0.3
3P20N5T	VRP-IM	128805	131262	1.9
	VRP-TIM	94850	101505	6.6
5P20N5T	VRP-IM	150075	165037	9.1
	VRP-TIM	96008	106671	10.0
1P30N5T	VRP-IM	213697	214955	0.6
	VRP-TIM	126989	127369	0.3
3P30N5T	VRP-IM	218268	229654	5.0
	VRP-TIM	149120	165583	9.9
5P30N5T	VRP-IM	342772	376994	9.1
	VRP-TIM	305375	311610	2.0
1P40N5T	VRP-IM	698592	713216	2.1
	VRP-TIM	555475	564037	1.5
3P40N5T	VRP-IM	1116978	1186197	5.8
	VRP-TIM	1113083	1136364	2.0
5P40N5T	VRP-IM	2207908	2278178	3.1
	VRP-TIM	1941518	1998841	2.9
Average		415526	431083	4.6

Table 5.7: Results of comparison between CPLEX and HGA on small and relatively large instances (T=5, P=[10;50])

Instance	Model	CPLEX (24 hours)	HGA	GAP (%)
		TC	TC	
10P10N5T	VRP-IM	140557	150529	6.6
	VRP-TIM	77430	79360	2.4
30P10N5T	VRP-IM	241628	244706	1.3
	VRP-TIM	117936	125302	5.9
50P10N5T	VRP-IM	270571	272303	0.6
	VRP-TIM	214298	219320	2.3
10P20N5T	VRP-IM	255809	266921	4.2
	VRP-TIM	166545	179466	7.2
30P20N5T	VRP-IM	357611	380396	6.0
	VRP-TIM	275612	295787	6.8
50P20N5T	VRP-IM	435518	480562	9.4
	VRP-TIM	283314	292944	3.3
10P30N5T	VRP-IM	545194	557441	2.2
	VRP-TIM	383061	402484	4.8
30P30N5T	VRP-IM	599030	643571	6.9
	VRP-TIM	436854	484457	9.8
50P30N5T	VRP-IM	1170652	1270142	7.8
	VRP-TIM	944584	973017	2.9
10P40N5T	VRP-IM	2477299	2488065	0.4
	VRP-TIM	1758354	1779353	1.2
30P40N5T	VRP-IM	4195039	4581517	8.4
	VRP-TIM	3091047	3223196	4.1
50P40N5T	VRP-IM	8994324	9160663	1.8
	VRP-TIM	5848865	5900509	0.9
Average		1386714	1435500	4.5

Table 5.8: Results for small and relatively large instances (T=3)

Instances	Model	Level	T	I	LS	Tr	TC	ϵ^* (%)	β^* (%)	SL (%)	CPU (s)
1P10N3T	VRP-IM	UL	2391	662	634	0	3688	-	95	47	42
		LL	0	3577	11288	0	14865				
	VRP-TIM	UL	2894	806	626	1433	5759	37	23	87	23
		LL	0	2445	187	842	3474				
3P10N3T	VRP-IM	UL	4623	1438	3347	0	9408	-	81	40	73
		LL	0	5769	14632	0	20402				
	VRP-TIM	UL	6292	1644	149	2105	10190	37	69	73	16
		LL	0	5569	330	1237	7136				
5P10N3T	VRP-IM	UL	7750	2651	4594	0	14994	-	62	43	19
		LL	0	9344	7520	0	16865				
	VRP-TIM	UL	9165	2592	178	1408	13343	40	28	74	11
		LL	0	7059	70	930	8059				
1P20N3T	VRP-IM	UL	7225	3477	5169	0	15871	-	60	48	55
		LL	0	11209	7806	0	19014				
	VRP-TIM	UL	7544	3090	0	1633	12266	51	0	100	19
		LL	0	7576	0	1689	9264				
3P20N3T	VRP-IM	UL	9004	4045	8553	0	21602	-	55	43	12
		LL	0	10800	10488	0	21288				
	VRP-TIM	UL	12441	3607	407	1285	17740	59	43	79	35
		LL	0	11064	302	1845	13210				
5P20N3T	VRP-IM	UL	13884	6379	15996	0	36260	-	39	35	79
		LL	0	11156	10359	0	21515				
	VRP-TIM	UL	13329	4863	25	2944	21160	48	96	87	76
		LL	0	10405	615	2675	13695				
1P30N3T	VRP-IM	UL	20608	7200	2139	0	29947	-	92	35	56
		LL	0	19500	25965	0	45466				
	VRP-TIM	UL	15996	6963	0	2733	25693	51	100	100	40
		LL	0	16010	0	2823	18833				
3P30N3T	VRP-IM	UL	17844	9891	16264	0	43998	-	58	46	12
		LL	0	18254	22250	0	40504				
	VRP-TIM	UL	27281	9063	106	4210	40660	40	18	88	15
		LL	0	18329	23	2753	21105				
5P30N3T	VRP-IM	UL	30478	14134	36659	0	81270	-	45	34	37
		LL	0	35832	29924	0	65756				
	VRP-TIM	UL	62146	19060	197	10028	91430	25	25	78	63
		LL	0	27109	66	3272	30447				
1P40N3T	VRP-IM	UL	72232	31999	119258	0	223489	-	15	48	78
		LL	0	55827	21209	0	77036				
	VRP-TIM	UL	101981	25808	111	13533	141432	49	70	87	45
		LL	0	58447	263	12973	71683				
3P40N3T	VRP-IM	UL	94046	53118	177614	0	324778	-	41	42	39
		LL	0	83517	124380	0	207897				
	VRP-TIM	UL	154705	35821	515	26084	217125	42	32	89	32
		LL	0	78670	241	18657	97568				
5P40N3T	VRP-IM	UL	155834	111866	300065	0	567764	-	50	33	74
		LL	0	173799	306018	0	479818				
	VRP-TIM	UL	273518	46603	343	77884	398348	19	67	86	47
		LL	0	147900	701	17996	166597				

Table 5.9: Results for small and relatively large instances (T=5)

Instances	Model	Level	T	I	LS	Tr	TC	ϵ^* (%)	β^* (%)	SL (%)	CPU (s)
1P10N5T	VRP-IM	UL	7612	2015	10471	0	20098	-	71	30	7
		LL	0	11678	25598	0	37276	-			
	VRP-TIM	UL	9086	2172	337	355	11950	94	85	80	79
		LL	0	7082	1910	5629	14621				
3P10N3T	VRP-IM	UL	14207	4075	7354	0	25636	-	85	39	63
		LL	0	19263	41910	0	61173	-			
	VRP-TIM	UL	15102	4524	1581	3319	24525	67	1	81	15
		LL	0	14569	18	6642	21228				
5P10N5T	VRP-IM	UL	22079	8933	7890	0	38903	-	79	44	95
		LL	0	31424	30344	0	61768	-			
	VRP-TIM	UL	30803	8046	123	6341	45313	13	84	88	12
		LL	0	21544	631	975	23150				
1P20N5T	VRP-IM	UL	18692	8900	30779	0	58372	-	5	36	93
		LL	0	29580	1598	0	31177	-			
	VRP-TIM	UL	24652	8438	0	4016	37106	58	92	100	17
		LL	0	20666	0	5564	26230				
3P20N5T	VRP-IM	UL	30244	10291	23375	0	63909	-	62	38	21
		LL	0	29527	37825	0	67352	-			
	VRP-TIM	UL	41367	10543	334	7031	59275	30	85	86	102
		LL	0	37186	1963	3081	42230				
5P20N5T	VRP-IM	UL	44179	21824	43414	0	109417	-	36	31	100
		LL	0	30846	24775	0	55621	-			
	VRP-TIM	UL	38240	13970	948	1900	55059	89	46	69	36
		LL	0	34711	800	16102	51613				
1P30N5T	VRP-IM	UL	49213	17920	36635	0	103769	-	60	39	10
		LL	0	56395	54791	0	111187	-			
	VRP-TIM	UL	40870	16635	0	6957	64462	54	15	100	110
		LL	0	54883	0	8023	62906				
3P30N5T	VRP-IM	UL	49232	26062	11793	0	87086	-	89	30	38
		LL	0	49121	93446	0	142567	-			
	VRP-TIM	UL	72349	25359	4	2742	100453	86	99	88	69
		LL	0	47326	338	17465	65130				
5P30N5T	VRP-IM	UL	69564	39692	18525	0	127781	-	89	31	46
		LL	0	91906	157307	0	249213	-			
	VRP-TIM	UL	141514	42578	552	19266	203910	49	28	87	15
		LL	0	89021	220	18459	107700				
1P40N5T	VRP-IM	UL	125424	95261	10848	0	231533	-	97	35	6
		LL	0	158078	323604	0	481682	-			
	VRP-TIM	UL	201091	73320	1662	31556	307629	59	0	81	80
		LL	0	211603	5	44800	256408				
3P40N5T	VRP-IM	UL	222753	220815	320036	0	763604	-	40	33	43
		LL	0	208980	213614	0	422594	-			
	VRP-TIM	UL	387703	156831	2185	64299	611018	62	14	83	54
		LL	0	420244	347	104754	525346				
5P40N5T	VRP-IM	UL	384250	526864	781004	0	1692117	-	26	47	94
		LL	0	318694	267366	0	586060	-			
	VRP-TIM	UL	588534	267396	1374	151755	1009059	45	66	77	14
		LL	0	865282	2727	121773	989782				

Table 5.10: Results for small and relatively large instances (T=5, P=[10;50])

Instances	Model	Level	T	I	LS	Tr	TC	ϵ^* (%)	β^* (%)	SL (%)	RT (s)
10P10N5T	VRP-IM	UL	21085	6526	39934	0	67545	-	53	38	223
		LL	0	37405	45579	0	82984				
	VRP-TIM	UL	28820	6914	2612	10674	49021	26	59	87	171
LL	0	22776	3792	3771	30339						
30P10N5T	VRP-IM	UL	42350	13659	100613	0	156622	-	28	39	304
		LL	0	49391	38693	0	88084				
	VRP-TIM	UL	35670	11757	3484	15742	66652	48	9	73	101
LL	0	43780	350	14519	58649						
50P10N5T	VRP-IM	UL	52725	20582	34759	0	108066	-	71	40	182
		LL	0	79723	84514	0	164237				
	VRP-TIM	UL	103990	23091	1464	9295	137840	47	36	67	69
LL	0	72432	807	8241	81481						
10P20N5T	VRP-IM	UL	57778	22188	58632	0	138598	-	43	33	196
		LL	0	84775	43547	0	128323				
	VRP-TIM	UL	60225	25533	0	8528	94286	64	94	74	82
LL	0	70079	0	15100	85180						
20P20N5T	VRP-IM	UL	98413	32446	93782	0	224642	-	41	44	107
		LL	0	89615	66140	0	155754				
	VRP-TIM	UL	134733	27643	1911	12107	176395	53	71	67	259
LL	0	100887	4775	13731	119392						
50P20N5T	VRP-IM	UL	104792	74289	215707	0	394789	-	0	34	311
		LL	0	85444	329	0	85773				
	VRP-TIM	UL	100074	42134	1672	31652	175532	32	62	87	239
LL	0	100071	2774	14567	117412						
10P30N5T	VRP-IM	UL	115159	49783	101081	0	266022	-	56	33	300
		LL	0	163546	127873	0	291419				
	VRP-TIM	UL	118974	56093	0	24236	199303	49	64	70	144
LL	0	179467	0	23714	203181						
30P30N5T	VRP-IM	UL	128100	72452	251249	0	451801	-	21	48	116
		LL	0	126537	65234	0	191770				
	VRP-TIM	UL	205036	75621	286	46096	327039	30	72	86	197
LL	0	136584	753	20081	157418						
50P30N5T	VRP-IM	UL	224176	172000	74357	0	470532	-	89	45	194
		LL	0	201573	598037	0	799609				
	VRP-TIM	UL	462151	182777	381	67978	713287	44	83	65	121
LL	0	203511	1904	54315	259730						
10P40N5T	VRP-IM	UL	477719	388892	737077	0	1603688	-	45	37	307
		LL	0	284218	600160	0	884378				
	VRP-TIM	UL	806453	420386	3503	238992	1469334	1	32	82	266
LL	0	306690	1646	1682	310018						
30P40N5T	VRP-IM	UL	995566	710506	1941002	0	3647074	-	16	42	312
		LL	0	575256	359187	0	934443				
	VRP-TIM	UL	1276615	733995	3685	142791	2157085	75	48	69	80
LL	0	636996	3434	425681	1066111						
50P40N5T	VRP-IM	UL	1713368	1617823	3293957	0	6625149	-	31	41	73
		LL	0	1084358	1451156	0	2535515				
	VRP-TIM	UL	2280034	1418812	1766	833662	4534273	38	83	67	143
LL	0	838923	8579	518733	1366236						

favour of or against any of the participants, regardless of their relative power and leadership position in the supply chain. This result indicates that the transshipment-related agreement between customers frequently needs to be accompanied by mutual monitoring and enforcement mechanisms and possibly pre-negotiated cash payments. Coordination is thus beneficial, i.e. encouraging the decision-makers to behave as in the centralised system while still optimising their income. One way of organising the system is to allow the decision-makers to transfer payments based on their actions and implication. Finally, as for HGA performance, we can see from Table 5.5, 5.6, and 5.7 that HGA can provide solutions with a minimum gap (on average 3.2%) and with less amount of time. However, we can notice that in some instances, the gap exceeds 5%—this highlight the limitation of GA in terms of scalability.

5.6 . Conclusions & perspectives

This paper considers a deterministic, multi-product, multi-period, multi-vehicle routing problem with transshipment; in a decentralised supply chain. The supply chain consists of a CW that produces a set of items and sells them through an independent network of customers, running on a franchising scheme. We model the problem as a BLP. We consider that the CW acts as a Stackelberg leader. That is, the lower levels: customers optimise their objective function subject to the value of the leader variable. The CW (UL) goal is to choose a stock level and routes to be constructed (according to its objective), knowing that the customers will follow optimally; while satisfying a set of constraints relative to vehicle routing, product availability and inventory management. The results show that if the CW agrees to take part in the costs of LT and lost sales, the network will improve its service level. They also show that the CW seems to become increasingly more involved as more transshipment and the customer incurs shortage costs. Furthermore, all of the parties involved can end up worse off once transshipment is permitted. Hence, depending on the value of the parameters ϵ^* and β^* , sharing of inventories may work for and against any of the parties, regardless of their relative power and leadership position in the supply chain. This result suggests that sharing deals where customers are autonomous also needs to be followed by joint control and regulation systems and likely pre-negotiated cash payments between the parties. Further, some supply chains are characterised not only by one or two decision-makers but also by many divisions at the lower level where there is a mix of small and large customers arranged within a hierarchical structure. The BLP we develop could also be generalised to model such decentralised systems. Finally, as for the resolution approach, coupling HGA with reinforcement learning helps significantly to enhance the quality of the feasible solutions and reduce the computational time. While the benefits of the model and the effectiveness of the resolution approach are demonstrated using randomly generated instances, it would be helpful to further assess their effectiveness on a real dataset. Also, it would be interesting to evaluate the performance and limitation of resolution approaches on other complex supply chains such as closed-loop ones.

6 - Modeling and solving a deterministic multi-vehicle IRP-T with pickup and delivery of shared RTIs

Preliminary version of the paper published on the proceedings of 9th International Conference on Operations Research and Enterprise Systems, ICORES 2020, VALLETA, Malta

Achamrah, F.E., Riane, F., Bouras., A, and Sahin, E.(2020). Collaboration Mechanism for Shared Returnable Transport Items in Closed Loop Supply Chains. 9th International Conference on Operations Research and Enterprise Systems, ICORES 2020, VALLETA, Malta. pp. 247-254

Full paper published in Sustainability Journal, ISSN 20711050, Impact Factor: 3.2, Quartile: Q1. <https://doi.org/10.3390/su14105805>

Achamrah, F.E., Riane, F., Sahin, E., and Limbourg, S. (2022). An Artificial-Immune-System-Based Algorithm Enhanced with Deep Reinforcement Learning for Solving Returnable Transport Item Problems. Sustainability. p.5805.

An Artificial-Immune-System-Based Algorithm Enhanced with Deep Reinforcement Learning for Solving Returnable Transport Item Problems

Fatima Ezzahra Achamrah ^{a,b}, Fouad Riane ^{a,b}, Evren Sahin ^b, Sabine Limbourg ^c,

^aComplex Systems and Interactions, Ecole Centrale of Casablanca; ^b Laboratoire Genie Industriel, CentraleSupélec, Paris Saclay University; ^c HEC-University of Liege (ULiege)

Abstract: This paper proposes a new approach, i.e., virtual pooling, for optimising returnable transport item (RTI) flows in a two-level closed-loop supply chain. The supply chain comprises a set of suppliers delivering their products loaded on RTIs to a set of customers. RTIs are of various types. The objective is to model a deterministic, multi-supplier, multi-customer inventory routing problem with pickup and delivery of multi-RTI. The model includes inventory-level constraints, the availability of empty RTIs to suppliers, and the minimisation of the total cost, including inventory holding, screening, maintenance, transportation, sharing, and purchasing costs for new RTIs. Furthermore, suppliers with common customers coordinate to virtually pool their inventory of empty RTIs held by customers so that, when loaded RTIs are delivered to customers, each may benefit from this visit to pick up the empty RTI, regardless of the ownership. To handle the combinatorial complexity of the model, a new artificial-immune-system-based algorithm coupled with deep reinforcement learning is proposed. The algorithm combines artificial immune systems' strong global search ability and a strong self-adaptability ability into a goal-driven performance enhanced by deep reinforcement learning, all tailored to the suggested mathematical model. Computational experiments on randomly generated instances highlight the performance of the proposed approach. From a managerial point of view, the results stress that this new approach allows for economies of scale and cost reduction at the level of all involved parties to about 40%. In addition, a sensitivity analysis on the unit cost of transportation and the procurement of new RTIs is conducted, highlighting the benefits and limits of the proposed model compared to dedicated and physical pooling modes.

Keywords: closed loop supply chain, returnable transport items, pickup and delivery, inventory routing problem, artificial immune systems, deep reinforcement learning

6.1 . Introduction

Returnable transport items (RTIs) are all reusable assets used to facilitate product shipping, storing, handling, and protection in the supply chain (Cobb, 2016). RTIs cover reusable drums, pallets, crates, rolls, boxes, and barrels (Kim and Glock, 2014; Limbourg et al., 2016; Iassinovskaia et al., 2017). Along with globalised supply chains, the use of RTIs has become more prevalent in recent decades as they eliminate the wastes that one-way secondary packaging may generate (Glock, 2017). The use of RTIs has been proved to be an enabler for better ergonomics and productivity while facilitating automation, better inventory control, and improved quality (Limbourg et al., 2016; Iassinovskaia et al., 2017; Bortolini et al., 2018; Liu et al., 2020). Furthermore, their operational benefits help reduce the disposal costs of packaging material and improve productivity (Twede and Clarke, 2005). These assets usually flow in a closed-loop supply chain between players (Glock, 2017; Sarkar et al., 2017). Loaded RTIs are received and unloaded at a given level of the supply chain. Either the empty RTI can be collected and returned to the sender,

or the receiver can reuse them to ship his products and thus continue to flow downstream the supply chain. Therefore, there exist two flows of RTIs that must be managed (Talaei et al., 2016): forward flows, which correspond to the forward distribution of goods loaded on RTIs, and reverse flows, which correspond to the collection and return of empty RTIs to their owners. This paper aims to optimise both forward and reverse flows of RTIs in a two-level closed supply chain.

Managing such assets has become a primary concern of supply chain managers, along with managing warehouses, machines, and vehicles (Sarkar et al., 2017; Liu et al., 2020). Indeed, it has become very pressing for companies to effectively package products and guarantee to have them in the proper quantity, at the right place, and at the right time. To avoid shortages, many companies frequently invest in more RTIs, resulting in higher holding and purchasing costs (Limbourg et al., 2016; Meherishi et al., 2021). Moreover, supply chain players experience RTI losses, varying from 3 to 20% (TrackX, 2017). This mismanagement lengthens turnaround times and pushes players to overinvest in these assets, leading to inefficient budgetary practices: companies buy new RTIs to replace the lost ones and recruit additional staff to handle them (Kim and Glock, 2014; Sarkar et al., 2017; TrackX, 2017).

According to Liu et al. (2020); Na et al. (2019); Zhang et al. (2015), RTI management can be divided into two modes depending on the ownership of empty RTIs: a dedicated mode (private RTIs) and a shared mode (public RTIs). The dedicated mode (DM) refers to the case where RTIs are owned by players (suppliers, for example) who use them exclusively to deliver their products without considering sharing them with others. They are responsible, in general, for collecting, refurbishing, and managing the inventory of their specific assets. In this system, RTIs received by a partner are shipped back to their specific owner. In the shared mode (SM), players agree to share their RTIs within a "pooled" system. A service provider company manages this shared system, and running such a pool is its core business (GS1 Global Office, 2014). In this pool, empty RTIs are physically stored and can be used by all players without any obligation for these assets to return to their starting point at their next movement (GS1 Global Office, 2014). RTI pools can be categorised into two types: "rented" and "open" pools (GS1 Global Office, 2014). The "rented" pool is based on a one-owner pool model: RTIs are owned by one company that rents and provides the supply chain players with the empty RTIs they need. In this case, the company manages and oversees its RTI pool's day-to-day operations and services. The "open" pool is based on a changing-owner pool model: all partners store their RTIs in a pool, and when an RTI is used, its ownership is transferred to the receiving partner, who must return similar RTIs of comparable quality (1:1 exchange concept). In both cases, a pooling system involves a pooler responsible for supplying ready-to-use RTIs to all partners, collecting them from downstream levels, refurbishing damaged ones, and holding inventory within its facilities until new RTI orders are placed (Accorsi et al., 2019).

The literature review (see Section 6.2) shows that most papers exclusively address DM and SM and highlight each mode's benefits on the overall supply chain performance. However, both modes may not always be profitable and practicable. Compared to the SM, the DM may be easier to implement, and it does not lead to resource dependency, as each player is always free to manage and use his inventory of empty RTIs (Liu et al., 2020). On the other hand, the SM is typically less expensive, as it may offer cost benefits through the shared use of RTIs among tier suppliers (Zhang et al., 2015; Na et al., 2019). However, the prerequisites of commonly serviceable RTIs for various materials from several suppliers are hard to meet (Na et al., 2019). Moreover, the SM compels advanced decision-making on where to locate pooler facilities, how to set facilities' capacities, and how to distribute transportation flows (i.e., delivery, pickup, inventory balancing, and supply) across the

network, which may imply additional managerial costs (i.e., transportation, inventory holding) and a need for solid information system support (Govindan et al., 2015). The SM may also establish a resource dependency, as each player is not always free to pick up the empty RTIs needed to deliver his products. This is particularly true for complex supply chains, which include multiple origins and destinations and multiple RTIs that flow within, in which constraints such as variable demands, vehicle capacity, and shortage are to be considered.

This paper proposes a new approach to overcome the shortcomings above of both modes in a closed-loop supply chain. Specifically, we consider the case of a two-level closed-loop supply chain comprising a set of suppliers delivering products to joint customers. We assume that each supplier owns RTIs that can be held in either his or the customer's inventory. In addition, each supplier is responsible, as in DM, for collecting, refurbishing, and managing his inventory. We also assume that the suppliers coordinate their logistics operations so that, while delivering loaded RTIs to customers, each supplier may benefit from this visit to pick up empty RTIs regardless of the ownership. This has earmarks of the classic lateral transshipment that relies on authorising the virtual pooling of finished products' inventory between members belonging to the same echelon of supply chains (Paterson et al., 2011). This practice usually takes place to re-balance the entire system's stock levels to react to scenarios where one of the locations faces a shortage while others have residual stock in hand. Accordingly, instead of calling upon a pooler or a leasing company to acquire the needed quantities of RTIs, this paper suggests that suppliers arrange to "virtually" pool/share their stock of identically substitutable RTIs: no need for a real and physical pool to store RTIs as in SM. As such, we conserve the ownership of RTIs as in the DM and allow the shared use of RTIs as in SM (Table 6.1). Moreover, each supplier buys, when needed, and adds new RTIs to the whole system. Therefore, the order may be filled, and the customers receive what they want, and the partners free up space in their inventory and reduce idle stock. It is mutually beneficial for all parties. Consequently, suppliers can sidestep the shortage of empty RTIs at their levels and reduce the cost of transportation, inventory holding, and the procurement of new RTIs. Such a strategy creates a valuable partnership but implies additional logistics operations that must be optimised.

Table 6.1: Characteristics of RTIs management strategies

	DM	SM	Virtual Pooling Mode
Owner of RTIs	Each supplier	All suppliers or a pooler company	Each supplier
Management of empty RTIs, collection, refurbishing	Each supplier	One pooler company	All suppliers
Storage of empty and shared RTIs	-	In dedicated facilities	At suppliers' level

Our paper has three main contributions. First, we develop a new mathematical formulation of the RTI pickup and delivery problem in a closed-loop supply chain consisting of a set of suppliers shipping their products to a set of common customers (e.g., plants, retailers) and using a set of RTIs, i.e., a multi-supplier multi-customer inventory routing problem with the pickup and delivery of multi-type of shared RTIs (IRPPDS). We assume that supply chain partners adopt a vendor-managed inventory policy (VMI): their operations are coordinated to organise deliveries and pickups to fulfil customers' demands. Thus, we address a multi-supplier, multi-customer, multi-RTI inventory routing problem that is hard to solve due to its inherent combinatorial complexity. Suggesting an efficient way to cope with this complexity by developing a breakthrough solving approach is the second contribution of this paper. Indeed, we use a matheuristic that hybridises an artificial-immune-system-based metaheuristic and a mathematical programming algorithm. Furthermore, thanks to its generality and flexibility, this matheuristic uses deep

reinforcement learning techniques initially proposed by [Achamrah et al. \(2021\)](#) for successfully solving dynamic and stochastic inventory routing problems. Also, the performance of the approach is compared to the one developed in [Achamrah et al. \(2021\)](#) and two pure metaheuristics. Finally, broad experiment campaigns are conducted on instances of large sizes. These experiments stress that the resolution approach is very competitive compared to other existent metaheuristics: it leads to better quality solutions and reduces computational time. Furthermore, we evaluate the cost reduction enabled by the virtual pooling of RTIs compared to DM and SM.

The remainder of the paper is organised as follows. Section 6.2 presents an overview of related works. After a detailed definition of the problem in Section 6.3, the mathematical formulation is provided in Section 6.4. Section 6.5 describes the proposed resolution approach and explains the hybridisation scheme used to integrate the mathematical model, the artificial-immune-system-based algorithm and the deep reinforcement learning technique. Section 6.6 provides the computational results and presents the metaheuristic performance analysis compared to three resolution approaches. Finally, Section 6.7 summarises the main findings and provides perspectives for further research.

6.2 . Related Work

This section reviews research streams that are mostly related to our work. The objective is to position our contributions in line with papers on the inventory routing problem (IRP) with pickup and delivery and RTIs management modes and highlight our contribution to the resolution approaches applied to solve similar problems.

The vehicle routing problem (VRP) calls for determining the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers ([Toth and Vigo, 2002](#)). In the literature, three different variants related to the structure of pickup and delivery and the number of origins and destinations are to be distinguished ([Berbeglia et al., 2007](#)): one-to-one (1-1), in which a request is originated at one location and destined for another location; one-to-many-to-one (1-M-1), in which each customer receives a delivery originating from a common depot and sends a pickup quantity to the depot; and finally, many-to-many (M-M), in which a commodity may be picked up at one of many locations and also delivered to one of many locations ([Andersson et al., 2011](#); [Rais et al., 2014](#); [Chen et al., 2014](#); [Li et al., 2015](#)). The IRP calls for inventory management, vehicle routing, and delivery scheduling decision-making problems ([Coelho and Laporte, 2014](#)). Our paper's most relevant research stream addresses IRP with pickup and delivery (IRPPD). According to [Parragh et al. \(2008\)](#), this problem has three variations regarding vehicle routing: (1) VRP with simultaneous pickup and delivery (SPD), in which products are delivered whilst others are simultaneously sent back to the origin; (2) VRP with back-hauls, where all deliveries must be undertaken before any pickup on each route; (3) VRP with mixed pickup and delivery, which can be characterised as a particular case of the VRP with SPD in which customers may have a pickup or delivery demands. Some recent applications of the VRP/IRP with pickups and deliveries can be found in ([Iassinovskaia et al., 2017](#); [Liu et al., 2020](#); [Tarantilis et al., 2012](#); [Archetti et al., 2018](#); [Van der Heide et al., 2018](#); [Archetti et al., 2020](#)). IRP problems have been intensively studied in the literature, and the reader is referred to ([Coelho et al., 2014b](#)) for a thorough overview of more related papers. Furthermore, for the more recent papers on decision support models for RTIs, the reader is referred to the review by ([Meherishi et al., 2019](#); [Glock, 2017](#)), which provides a systematic literature review of decision models in managing closed-loop supply chains, including RTIs. Along with developing decision support models, significant research efforts have also been devoted to investigating RTI management strategies in both

the dedicated and the shared modes (Zhang et al., 2015). Most related works address the management of RTIs as part of a VMI policy and develop decision support models for cost reduction under stochastic or deterministic environments for the dedicated mode. Applications can be found in (Cobb, 2016; Kim and Glock, 2014; Zhang et al., 2015). In Achamrah et al. (2019); Iassinovskaia et al. (2017); Singh et al. (2017), the authors propose models for inventory routing problems with simultaneous pickups and deliveries for a single-supplier, single-RTI, multi-customer (1-M-1) closed-loop supply chain. The models consider the maintenance costs of the reused RTIs returned from customers and the cost of buying a new one. In Achamrah et al. (2019), other scheduled pickups and the supply of new RTIs are integrated as alternatives to sidestep the shortage of empty RTIs at the supplier level. Finally, in Meherishi et al. (2021), a decentralised two-stage supply chain with a Retailer Stackelberg game is studied. The authors develop an analytical model to determine lot-sizing and pricing decisions for the product and its secondary packaging. As for the shared mode, most related work has studied different scenarios for the pooling or rental of RTIs with the help of mathematical modelling and simulation. The authors of Ech-Charrat et al. (2017) investigate a lot-sizing problem and assignment strategy that minimises the pallet management cost under environmental constraints. The authors of Ren et al. (2017) study the pallet allocation problem under stochastic supply scenarios and customer priority. In contrast, those of Bortolini et al. (2018) study a fresh fruit and vegetable supply chain and develop a mathematical model to select the best packaging (reusable/disposable) and minimise holding and handling costs. The authors of Tornese et al. (2018) analyse the effects of pallet service conditions and repair facilities on a pallet pooling system's economic and environmental performance. In their paper, a new RTI procurement decision is also taken into consideration. The authors of Hassanzadeh Amin et al. (2018) analyse the reverse logistics of plastic pallets in Canada, focusing on recovery options, such as reusing, remanufacturing, and recycling. A mathematical model is developed to determine the best locations in a pallet reverse logistics network and optimise the distribution flows between the network players. The authors of Accorsi et al. (2019) analyse the transportation operations of a pallet pooling company serving a set of retailers. A pooler company is assumed to be responsible for supplying, collecting, and refurbishing pallets. Buying/selling and pooling management strategies are assessed and compared through what-if analysis. The authors of Zhou and Song (2019) study the service centres' location problem considering a pallet pool mode. By integrating the forward and reverse flow of pallets, the objective is to minimise the total cost, including fixed construction, inventory, delivery, and recovery costs. The authors of Liu et al. (2020) develop a mixed-integer program model for planning the distribution and vehicle routing for a single type of RTI and in a single period. They consider a pooler company responsible for dispatching leased empty containers to its customers and collecting the customers' surplus empty containers. In their model, minimising procurement, storage, and maintenance costs are not considered. The authors of Achamrah et al. (2020) use a simulation-based approach to model sharing a single RTI between two producers in a closed-loop supply chain. The results show that collaboration can lead to economies of scale and cost reduction. They also highlight the need for a third party to manage the system to promise mutual benefits for the concerned parties. On the other hand, the routing decisions are not optimised in their simulation model. Moreover, the model is not generic and realistic, as it considers a simple supply chain and only one type of RTI that flows in.

As for combinatorial complexity, VRP/IRP with pickup and delivery problems are well known to be NP-hard (Karp, 1972; Papadimitriou and Steiglitz, 1977). To tackle this complexity, approximation algorithms or metaheuristics are used. The most commonly encountered metaheuristics are either stochastic algorithms such as simulated annealing (SA) or ones based on artificial intelligence algorithms such as artificial immune system

(AIS), genetic algorithm (GA), particle swarm optimisation (PSO) and ant colony (AC). Though AIS-based algorithms are a relatively new complex-problem-solving approach compared to other metaheuristics, the inherent characteristics of the immune memory, vaccination process, and self-recognition ability of the antibody and the diversity of immunity allow it to have a high level of flexibility and a good balance between global and local search (Bernardino and Barbosa, 2009b). Furthermore, AIS has demonstrated efficiency in convergence compared to other algorithms for large instances. The authors of De Castro et al. (2000); Wong et al. (2010); Tiwari et al. (2005); Panigrahi et al. (2007); Pierrard and Coello Coello (2012); Navarro et al. (2018) reported that AIS has a higher convergence rate than GA, PSO, AC, and SA. Therefore, AIS is used to solve our model for large-sized instances for all the reasons above.

To further enhance the convergence speed of AIS, we use machine learning (ML) techniques. Indeed, metaheuristics, through their iterative search processes, generate a lot of data that can be turned into explicit knowledge if coupled with ML models. This data concerns decision-making solutions and the objective spaces visited during the search process, the sequence of solutions or trajectories, successive populations of solutions, moves, recombination, local optima, elite solutions, and bad solutions (Talbi, 2020). ML techniques can help analyse this data, extract valuable knowledge, and enhance metaheuristics' search performance. Thus, metasearch techniques become "data-driven", "well informed", and therefore "smarter". In this respect, ML was used to address discrete optimisation problems that focus on the travelling salesman problem and VRP. The data-driven metaheuristics have been proven to be advantageous in convergence speed, solution quality, and robustness. The methodologies in ML for decision problems, typically addressed by operation research (OR), are mainly found in reinforcement learning (RL), learning to search, and multi-armed bandits. The authors of (Bello et al., 2016; Dai et al., 2017; Han et al., 2016; Kaempfer and Wolf, 2018; Achamrah et al., 2021) illustrate the recent successes achieved by RL concerning problems typically addressed by OR. For instance, the authors of (Achamrah et al., 2021) develop a metaheuristic enhanced by RL techniques to solve a dynamic and stochastic IRP. The authors of (Ahmadian, 2017; OroojlooyJadid and Hajjinezhad, 2019) introduce ML in the solution processes of inventory and location problems. Finally, the authors of (Lu et al., 2019; Duan et al., 2020; Achamrah et al., 2022a) use an RL-based technique to solve a VRP. As far as we are concerned, our paper is the first that combines RL with AIS to solve a multi-supplier multi-customer multi-RTI IRP with pickup and delivery in a closed-loop supply chain (Nakib et al., 2017; Seyyedabbasi et al., 2021).

This review shows that despite the extensive literature on RTIs related to IRP, there is a lack of efficient tools and techniques to solve complex combinatorial problems such as closed-loop multi-product, multi-period, inventory routing problems with deliveries and pickups of multiple types of RTIs. As already mentioned, our research makes three main scientific contributions. Firstly, we develop a mathematical model to address the deterministic, multi-supplier, multi-customer (M-M) inventory routing problem, considering the delivery and return flows of multiple RTIs which are virtually pooled between a given number of suppliers. Secondly, we use a new artificial immune-system-based algorithm and combine its strong global search capability with RL's strong self-adaptability and goal-driven performance, all tailored to the mathematical model. Thirdly, computational experiments on specially designed instances highlight the performance of the proposed algorithm. From a managerial point of view, the results stress that this new approach allows for economies of scale and cost reduction at the level of all the involved parties. Furthermore, a sensitivity analysis on unit cost and the procurement of new RTIs is conducted and highlights the benefits and limits of the proposed model compared to other RTI management modes.

6.3 . Mathematical Formulation

This section presents the mathematical models developed for IRPPDS, DM, and SM.

6.3.1 . Mathematical Model for IRPPDS

We examine a multi-supplier, multi-customer, multi-RTI closed-loop supply chain. A set of m suppliers distribute different types of products using a set of r types of RTIs to a set of n common customers over a finite planning horizon. Each supplier delivers RTIs loaded with products to a set of customers. Each customer uses these products in his production process and constitutes an inventory of empty RTIs. The supplier then collects those empty RTIs to be reused for future productions and deliveries at his level. We assume that all supply chain players adopt a centralised management policy to synchronise operations according to each player's requirements, optimise deliveries and pickups, and meet customers' expectations.

The planning horizon is defined by a discrete and finite set of periods (days). Each player has a storage zone separated into two areas: one for the inventory of empty RTIs (E) and another for the inventory of loaded RTIs (L). Each of these inventory areas is characterised by an initial inventory level and a maximum holding capacity. Initial inventories of loaded and empty RTIs are supposed to be positive and known at the beginning of the planning horizon. Deliveries and pickups are carried out by a set of homogeneous fleets of vehicles. Each vehicle can transport loaded or empty RTIs, or both, with a determined capacity in terms of the number of RTIs without distinction between empty and loaded RTIs (foldable RTIs are not considered). It is assumed that each constructed route starts from a supplier to visit a set of customers, and there is no route built between suppliers. Furthermore, customers are visited by each supplier independently of other suppliers' planned routes. Since vehicles have a limited capacity, multiple suppliers' routes are allowed. We assume that a vehicle can perform at most one pickup and delivery per period, all routes start and finish at each supplier, and split pickup/deliveries are not allowed. In each period, the sequence of events is as follows. First, each supplier prepares the quantity of loaded RTIs to be shipped by considering the current inventory. He uses his empty RTIs and those of other suppliers to load products on the appropriate type of RTIs. Then, each supplier visits each customer in each period to deliver the required quantity of products (in terms of loaded RTIs) for production. The available inventory of empty RTIs at each level of the supply chain is checked. Depending on the demand that he must satisfy in the next period, each supplier picks up empty RTIs belonging to him. If these are not sufficient; he picks up other RTIs belonging to the other suppliers as long as these latter have enough inventory to meet demands for the next period. After pickups are performed, the empty RTIs are subject to quality control at each supplier location. Damaged RTIs are disposed of; serviceable RTIs are repaired, and undamaged RTIs are transferred to the inventory of empty RTIs. All the RTIs present in the inventory (repaired/cleaned) at the end of each period can be reused in the next period. Moreover, we assume that, in addition to the virtual pooling of empty RTIs, a supplier can purchase empty RTIs that he may need to fulfil future demands. In this case, buying RTIs is permitted in each period, and each RTI is available for use in the following one.

The objective of the IRPPDS model is to determine, for each level of the supply chain and over the finite planning horizon, the quantity of loaded RTIs to be delivered by each supplier and the quantity of empty RTIs to be picked up by each customer and shared. The demand is supposed to be deterministic by being time-varying. Such planning considers the inventory-level constraints (no shortages, backlogs, or overstocking are allowed), the availability of empty RTIs to suppliers, and the minimisation of the total cost, including inventory holding, maintenance, transportation, sharing, and purchasing

of new RTIs.

To model IRPPDS, we introduce different notations. We consider: a set $N = \{i | i = 1, \dots, n\}$ of n customers; a set $P = \{0_p | 0_p = 0_1, \dots, 0_m\}$ of m suppliers; a set $N_p = \{i | i = 0_p, 1, \dots, n\}$ that represents the n customers, and the node 0_p that represents the supplier p ; a set $R = \{r | r = 1, \dots, u\}$ of u types of RTIs that are used to carry on different types of products; and a set $V = \{v | v = 1, \dots, k\}$ of k homogeneous vehicles with a capacity of Q in terms of the number of RTIs. Accordingly, loaded and empty RTIs occupy the same volume as in the case of boxes and containers. We also consider a horizon $T = \{t | t = 1, \dots, l\}$ of l periods. Each supplier p and customer i incurs a holding cost for loaded RTIs (L) and empty RTIs (E): $H_p^{L,r}, h_i^{L,r}, H_p^{E,r}$ and $h_i^{E,r}$ (€ per unit), respectively. $I_{p0}^{L,r}, L_{i0}^{L,r}, I_{p0}^{E,r}$ and $L_{i0}^{E,r}$ represent the initial inventory level of loaded and empty RTI of type r , respectively, at the supplier p and customer i . C_p^L, c_i^L, C_p^E and c_i^E represent the maximum holding capacity for loaded and empty RTI, respectively, for the supplier p and customer i . At the beginning of the planning horizon, each supplier p receives information on demand to satisfy D_{pit}^r (expressed in terms of loaded RTIs) of each customer $i \in N$ for each period $t \in T$ and for each RTI r . The distance between actors $i \in N_p$ and $j \in N_p$ is denoted by d_{ij}^p . The fixed cost of transportation is represented by a in € per km, and b represents the variable cost of transportation in € per weight unit and per km. The weights of a loaded and empty RTI are w_L^r and w_E^r , respectively. The cost of buying an RTI is e_r in € per unit. The sharing cost incurred by each supplier p is s_r per unit of unowned empty RTIs of type r belonging to other suppliers p used at his level to deliver products. This cost represents the utilisation cost of an unowned RTI used by each supplier if it occurs. Finally, g_r is the maintenance cost per RTI of type r used by the suppliers to deliver products, including inspection and cleaning costs. The model's notation is summarised in Table 6.2.

Table 6.2: Model's notation summary

Sets	
N	Set of n customers.
P	Set of m suppliers.
R	Set of u RTIs.
V	Set of k vehicles.
T	Set of l periods.
Parameters	
a	Fixed cost of transportation (€ per km).
b	Variable cost of transportation (€ per weight per km).
$H_p^{L,r}, h_i^{L,r}, H_p^{E,r}, H_i^{E,r}$	Cost of holding inventory of loaded and empty RTIs, respectively, for each supplier p and customer i .
e_r	Cost of buying a new RTI of type r (€ per unit).
s_r	Cost of sharing incurred by each supplier which is proportional to the number of unowned empty RTIs of a type r used at his level to deliver products (€ per unit of unowned RTI used).
g_r	Cost of maintenance of one RTI of type r (€ per RTI loaded).
w_L^r, w_E^r	Weights of a loaded and empty RTIs of type r , respectively.
Q	Capacity of vehicle in terms of the number of RTIs.
d_{ij}^p	Distance between nodes i and $j \in N_p$.
D_{pit}^r	Demand of each customer i for each period t loaded on an RTI r satisfied by supplier p .
$I_{p0}^{L,r}, L_{i0}^{L,r}, I_{p0}^{E,r}, L_{i0}^{E,r}$	Initial inventory level of loaded and empty RTIs of type r , respectively, for the supplier p and customer i .
$C_p^L, c_i^L, C_p^E, c_i^E$	Maximum holding capacity for loaded and empty RTIs, respectively, for the supplier p and customer i .
Decision variables	
x_{ijvt}^p	Binary variable stating whether the vehicle v visited node j immediately after node i in period t .
$F_{pt}^{p'}$	Quantity of empty RTIs of type r owned by supplier p that have been filled with products by supplier p' in period t . This quantity also includes the case where $p = p'$ (supplier uses his RTI).
I_{Lr}^{pt}	Inventory level of loaded RTIs of type r held by the supplier p at the end of period t .
L_{pit}^{Lr}	Inventory level of RTIs of type r filled with the product of supplier p by customer i at the end of period t .
$Q_{pit}^{p'r}$	Quantity of loaded RTIs of type r owned by supplier p' and delivered by supplier p to customer i in period t .
X_{ijt}^{pr}	Quantity of loaded RTIs of type r filled with a product of supplier p transported from node i to node j in period t .
L_{it}^{Er}	Inventory level of empty RTIs of type r held by the customer i at the end of period t .
I_{pt}^{Er}	Total quantities of all empty RTIs of type r held by the supplier p at the end of period t .
Z_{it}^{pr}	Quantity of empty RTIs of type r owned by supplier p collected from a customer i in period t .
$W_{ip't}^{pr}$	Quantity of empty RTIs of type r owned by supplier p and collected from customer i by supplier p in period t .
E_{ijt}^{pr}	Quantity of empty RTIs of type r collected by supplier p transported from node i to node j in period t .
n_t^{pr}	Quantity of RTIs of type r bought by supplier p in period t .

The IRPPDS in a multi-supplier, multi-customer, multi-RTI closed-loop supply chain

is modelled as follows:

$$\begin{aligned}
& \min \sum_{i \in N} \sum_{t \in T} \sum_{r \in R} (h_i^{Lr} L_{it}^{Lr} + h_i^{Er} L_{it}^{Er}) + \sum_{p \in P} \sum_{t \in T} \sum_{r \in R} (H_p^{L,r} I_{pt}^{Lr} + H_p^{Er} I_{pt}^{Er}) + \\
& \sum_{p \in P} \sum_{t \in T} \sum_{r \in R} e_r n_t^{p,r} + \sum_{p \in P} \sum_{t \in T} \sum_{p' \in P} \sum_{r \in R} g_r F_{pt}^{p'r} + \\
& \sum_{i \in N} \sum_{p \in P} \sum_{p' \in P} \sum_{t \in T} \sum_{r \in R} s_r W_{ip't}^{pr} + \\
& \sum_{p \in P} \sum_{t \in T} \sum_{i \in N_p} \sum_{j \in N_p} \left(a \sum_{v \in V} x_{ijvt}^p + \sum_{r \in R} b (w_L^r X_{ijt}^{pr} + w_E^r E_{ijt}^{pr}) d_{ij}^p \right)
\end{aligned} \tag{6.1}$$

subject to:

$$L_{pit}^{Lr} = L_{pit-1}^{Lr} + \sum_{p' \in P} Q_{pit}^{p'r} - D_{pit}^r \quad \forall i \in N, t \in T, p \in P, r \in R \tag{6.2}$$

$$I_{pt}^{Lr} = I_{pt-1}^{Lr} - \sum_{i \in N} \sum_{p' \in P} Q_{pit}^{p'r} + \sum_{p' \in P} F_{pt}^{p'r} \quad \forall t \in T, p \in P, r \in R \tag{6.3}$$

$$L_{it}^{Er} = L_{it-1}^{Er} - \sum_{p \in P} Z_{it}^{pr} + \sum_{p \in P} D_{pit}^r - \sum_{p \in P} \sum_{p' \in P} W_{ip't}^{pr} \quad \forall i \in N, t \in T, r \in R \tag{6.4}$$

$$I_{pt}^{Er} = I_{pt-1}^{Er} + \sum_{i \in N} Z_{it}^{pr} - \sum_{p' \in P} F_{pt}^{p'r} + n_t^{pr} + \sum_{p' \in P} W_{ip't}^{pr} \quad \forall p \in P, t \in T, r \in R \tag{6.5}$$

$$\sum_{i \in N_p, i \neq j} (X_{ijt}^{pr} - X_{jit}^{pr}) = \sum_{p' \in P} Q_{pjt}^{p'r} \quad \forall j \in N, p \in P, t \in T, r \in R \tag{6.6}$$

$$\sum_{i \in N_p, i \neq j} (E_{jit}^{pr} - E_{ijt}^{pr}) = Z_{jt}^{pr} + \sum_{p' \in P} W_{jp't}^{p'r} \quad \forall j \in N, p \in P, t \in T, r \in R \tag{6.7}$$

$$0 \leq \sum_{p \in P} \sum_{r \in R} L_{pit}^{Lr} \leq c_i^L \quad \forall i \in N, t \in T \tag{6.8}$$

$$0 \leq \sum_{r \in R} I_{pt}^{Lr} \leq C_p^L \quad \forall p \in P, t \in T \tag{6.9}$$

$$0 \leq \sum_{p \in P} \sum_{r \in R} L_{pit}^{Er} \leq c_i^E \quad \forall i \in N, t \in T \tag{6.10}$$

$$0 \leq \sum_{r \in R} I_{pt}^{Er} \leq C_p^E \quad \forall p \in P, t \in T \quad (6.11)$$

$$\sum_{p \in P} \sum_{r \in R} (X_{ijt}^{pr} + E_{ijt}^{pr}) \leq Q \sum_{p \in P} \sum_{v \in V} x_{ijvt}^p \quad \forall i, j \in N_p, t \in T \quad (6.12)$$

$$\sum_{i \in N_p} \sum_{v \in V} x_{ijvt}^p \leq 1 \quad \forall j \in N, p \in P, t \in T \quad (6.13)$$

$$\sum_{i \in N_p, i \neq j} x_{ijvt}^p = \sum_{i \in N_p, i \neq j} x_{jivt}^p \quad \forall v, s. \in V, j \in N_p, p \in P, t \in T \quad (6.14)$$

$$\sum_{j \in N} x_{0pjvt}^p \leq 1 \quad \forall v, s. \in V, p \in P, t \in T \quad (6.15)$$

The objective function (6.1) minimises inventory costs at the level of each customer and supplier, the costs of purchasing new RTIs, the cost of the maintenance of RTIs, the sharing cost of RTIs undertaken by each supplier, and finally, the fixed and variable cost of transportation for pickup and delivery. Constraints (6.2) define the conditions for the conservation of the inventory levels of loaded RTIs owned by supplier p at the level of each customer i . Constraints (6.3) state that at the level of each supplier p , the inventory level of loaded RTIs at the end of period t is equal to the inventory level at the beginning of the period minus the quantities of loaded RTIs delivered to all customers and the quantities of empty RTIs that were loaded by supplier p in period t . Constraints (6.4) indicate that the inventory level for customer i at the end of period t of empty RTIs, held by supplier p , is equal to the inventory level of empty RTIs at the beginning of the period minus the quantity picked up by each supplier p plus the RTIs that have been emptied after demand has been satisfied minus the quantity of empty RTIs belonging to each supplier p that other suppliers have collected. Constraints (6.5) indicate that at the level of each supplier p , the inventory level of empty RTIs at the end of period t is equal to the inventory level at the beginning of the period plus the quantity of his empty RTIs collected from all customers plus the quantity of empty RTIs belonging to other suppliers that have been collected from customers by supplier p , minus the quantity of empty RTIs that have been loaded in period t plus the quantity of purchased RTIs. Constraints (6.6) ensure that the quantities of loaded RTIs owned by supplier p are delivered to customer j . Constraints (6.7) show that the flow of empty RTIs belonging to supplier p outgoing from node j is equal to the quantity of empty RTIs belonging to supplier p collected by supplier p , plus the quantity of empty RTIs belonging to other suppliers collected by supplier p , minus the inflow from all customers. Constraints (6.8)–(6.11) indicate the boundaries of the inventory levels of loaded and empty RTIs at the level of each supplier p and customer i . Constraints (6.12) stipulate that the quantities delivered and collected

between two nodes i and j must not exceed the capacity of the vehicles on the arc (i, j) . Constraints (6.13)–(6.15) express the conditions for the construction of tours. Constraints (6.13) indicate that at most one vehicle is used to visit node j . Constraints (6.14) guarantee the continuity of a tour. Constraints (6.15) ensure that vehicles leave the supplier only once per period or remain at the depot. Finally, constraints that define the non-negative constraints and the binary nature of the decision variables are imposed.

6.3.2 . Mathematical Model for DM

For the DM model, there is no pooling of empty RTIs between the suppliers. That is, $W_{ip't}^{pr} = F_{pt}^{p'r} = 0, \text{ if } p' \neq p, \forall p, p' \in P, i \in N, t \in T, r \in R$. Each supplier manages, independently of other suppliers, the deliveries of his loaded RTIs to customers, the pickups of empty ones from customers, and their inventories at his level and customers' location. Accordingly, the mathematical model is solved for each supplier independently and costs to minimise include inventory holding of empty and loaded RTIs, the transportation cost for delivery and pickups, maintenance, and the procurement of new RTIs. Accordingly, the formulation of the DM model, $\forall p \in P$, is as follows:

$$\begin{aligned} \min & \sum_{i \in N} \sum_{t \in T} \sum_{r \in R} (h_i^{Lr} L_{it}^{Lr} + h_i^{Er} L_{it}^{Er}) + \sum_{t \in T} \sum_{r \in R} (H_p^{L,r} I_{pt}^{Lr} + H_p^{Er} I_{pt}^{Er}) + \\ & \sum_{t \in T} \sum_{r \in R} e_r n_t^{p,r} + \sum_{t \in T} \sum_{r \in R} g_r F_{pt}^r + \\ & \sum_{t \in T} \sum_{i \in N_p} \sum_{j \in N_p} \left(a \sum_{v \in V} x_{ijvt}^p + \sum_{r \in R} b (w_L^r X_{ijt}^{pr} + w_E^r E_{ijt}^{pr}) d_{ij}^p \right) \end{aligned} \quad (6.16)$$

The objective function minimises inventory costs for the supplier p and each customer, the costs of purchasing new RTIs, the maintenance cost of RTIs, and finally, the fixed and variable transportation costs for pickup and delivery.

It is subject to:

$$L_{pit}^{Lr} = L_{pit-1}^{Lr} + Q_{pit}^r - D_{pit}^r \quad \forall i \in N, t \in T, r \in R \quad (6.17)$$

$$I_{pt}^{Lr} = I_{pt-1}^{Lr} - \sum_{i \in N} Q_{pit}^r + F_{pt}^r \quad \forall t \in T, r \in R \quad (6.18)$$

$$L_{it}^{Er} = L_{it-1}^{Er} - \sum_{p \in P} Z_{it}^{pr} + \sum_{p \in P} D_{pit}^r \quad \forall i \in N, t \in T, r \in R \quad (6.19)$$

$$I_{pt}^{Er} = I_{pt-1}^{Er} + \sum_{i \in N} Z_{it}^{pr} - F_{pt}^r + n_t^{pr} \quad \forall t \in T, r \in R \quad (6.20)$$

$$\sum_{i \in N_p, i \neq j} (X_{ijt}^{pr} - X_{jit}^{pr}) = Q_{pj}^r \quad \forall j \in N, t \in T, r \in R \quad (6.21)$$

$$\sum_{i \in N_p, i \neq j} (E_{jit}^{pr} - E_{ijt}^{pr}) = Z_{jt}^{pr} \quad \forall j \in N, t \in T, r \in R \quad (6.22)$$

$$0 \leq \sum_{p \in P} \sum_{r \in R} L_{pit}^{Lr} \leq c_i^L \quad \forall i \in N, t \in T \quad (6.23)$$

$$0 \leq \sum_{r \in R} I_{pt}^{Lr} \leq C_p^L \quad \forall t \in T \quad (6.24)$$

$$0 \leq \sum_{p \in P} \sum_{r \in R} L_{pit}^{Er} \leq c_i^E \quad \forall i \in N, t \in T \quad (6.25)$$

$$0 \leq \sum_{r \in R} I_{pt}^{Er} \leq C_p^E \quad \forall t \in T \quad (6.26)$$

$$\sum_{p \in P} \sum_{r \in R} (X_{ijt}^{pr} + E_{ijt}^{pr}) \leq Q \sum_{p \in P} \sum_{v \in V} x_{ijvt}^p \quad \forall i, j \in N_p, t \in T \quad (6.27)$$

$$\sum_{i \in N_p} \sum_{v \in V} x_{ijvt}^p \leq 1 \quad \forall j \in N, t \in T \quad (6.28)$$

$$\sum_{i \in N_p, i \neq j} x_{ijvt}^p = \sum_{i \in N_p, i \neq j} x_{jivt}^p \quad \forall v, s. \in V, j \in N_p, t \in T \quad (6.29)$$

$$\sum_{j \in N} x_{0pjvt}^p \leq 1 \quad \forall v, s. \in V, t \in T \quad (6.30)$$

with:

Q_{pit}^r : Quantity of loaded RTIs of type r owned by supplier p and that have been delivered to customer i in period t .

F_{pt}^r : Quantity of empty RTIs of type r owned by supplier p and that have been filled with products at his level in period t .

6.3.3 . Mathematical Model for SM

In the SM model, a pooler company manages the inventory, pickups, and procurement of empty RTIs. On the other hand, each supplier is responsible for delivering loaded RTIs and managing their corresponding inventory. Furthermore, empty RTIs are delivered directly from customers to a series of centres (pooler facilities) managed by the company rather than suppliers, as in the DM and IRPPDS models. The centres are assumed to be located near the suppliers. To determine the location of these centres, we solve a multi-period weighted clustering problem (MPC). The clustering consists of grouping supplier nodes into clusters to minimise the distance between suppliers. Each cluster centroid of suppliers represents the centre in which empty RTIs of these suppliers are stored, cleaned, and repaired. When needed, the centre sends empty RTIs to suppliers so that they can produce and deliver their products to customers. As for costs, two other costs are to be considered: inventory holding at each centre ι and pooling cost. The latter incorporates the management of centres by the pooler company and each unowned RTI used by each supplier (which is assumed, for the purposes of simplification, to be equivalent to the sharing cost in IRPPDS). The constraints of IRPPDS for the inventory and routing of pickups of empty RTIs from customers to centres and from the centres to suppliers are rewritten accordingly. In the following, the formulation of the SM model is presented.

6.3.3.1 . Multi-Period Clustering Problem

To determine the location of the centres, we first solve an MPC. To do so, we define the binary variables $\theta_{p\iota}$ that have a value of 1 if a supplier p belongs to the cluster ι ($\iota \in K = \{\iota \mid \iota = 1, \dots, \kappa \leq m\}$), and 0 otherwise with a binary variable $\epsilon_{pp'\iota}$ having a value of 1 if the suppliers p and p' belong to the same cluster. MPC can be then modelled as follows:

$$\min \sum_{p,p' \in P} \sum_{\iota \in K} d_{pp'} \epsilon_{pp'\iota} \quad (6.31)$$

subject to:

$$\sum_{\iota \in K} \theta_{p\iota} = 1 \quad \forall p \in P \quad (6.32)$$

$$\sum_{p \in P} \sum_{r \in R} \sum_{1 \leq t' \leq t} D_{pit'}^r \theta_{p\iota} \leq tQ \quad \forall \iota \in K, t \in T \quad (6.33)$$

$$\epsilon_{pp'\iota} \leq \theta_{p\iota}, \epsilon_{pp'\iota} \leq \theta_{p'\iota} \quad \forall \iota \in K, p \in P, p' \in P, p \neq p' \quad (6.34)$$

$$\epsilon_{pp'\iota} \geq \theta_{p\iota} + \theta_{p'\iota} - 1 \quad \forall \iota \in K, p \in P, p' \in P, p \neq p' \quad (6.35)$$

$$\epsilon_{pp'\iota}, \theta_{p\iota} \in \{0, 1\} \quad \forall \iota \in K, p \in P, p' \in P, p \neq p' \quad (6.36)$$

The objective function (6.31) is to minimise the distance between suppliers (p, p') belonging to the same cluster (ι). Constraints (6.32) ensure that each supplier is assigned to a unique cluster. Constraints (6.33) state that the aggregate quantity of empty RTIs in each cluster in terms of demands over the planning horizon must fit into the available capacity, tQ , where Q is the vehicle capacity. Constraints (6.34) and (6.35) state that the distance between suppliers p, p' , and $d_{pp'}$ is included in the objective function if and only if suppliers p and p' are assigned to the same cluster. Constraints (6.36) define the binary nature of the decision variables.

6.3.3.2 . SM Model

In the SM model, two costs are considered:

- Inventory holding at each centre ι : $\sum_{\iota \in K} \sum_{t \in T} \sum_{r \in R} H_{\iota}^{Er} \mathcal{L}_{\iota t}^{Er}$;
- Pooling cost for each un-owned RTI used by each supplier (which is equivalent to the sharing cost in IRPPDS).

The constraints of IRPPDS for the inventory and routing of pickups of empty RTIs from customers to centres and from the centres to suppliers are rewritten as follows. Conservation of inventory levels and flows of empty RTIs at the level of each supplier p , customer i , and centre ι (with $\theta_{p\iota}$ and $\epsilon_{pp'\iota}$ already determined by solving MPC):

$$L_{it}^{Er} = L_{it-1}^{Er} - \sum_{\iota \in K} \sum_{p \in P} \theta_{p\iota} Z_{it}^{pr} + \sum_{\iota \in K} \sum_{p \in P} \theta_{p\iota} D_{pit}^r \quad \forall i \in N, t \in T, r \in R \quad (6.37)$$

$$I_{pt}^{Er} = I_{pt-1}^{Er} + \sum_{\iota \in K} \theta_{p\iota} R_{pt}^{\iota r} + \sum_{p' \in P} \sum_{i \in N} \sum_{\iota \in K} \epsilon_{pp'\iota} W_{ip't}^{pr} - \sum_{p' \in P} \sum_{\iota \in K} \epsilon_{pp'\iota} F_{pt}^{p'r} \quad \forall p \in P, t \in T, r \in R \quad (6.38)$$

$$\mathcal{L}_{it}^{Er} = \mathcal{L}_{it-1}^{Er} + \sum_{i \in N} \sum_{p \in P} \theta_{p\iota} Z_{it}^{pr} - \sum_{p \in P} \theta_{p\iota} R_{pt}^{\iota r} - \sum_{p, p' \in P} \sum_{i \in N} \epsilon_{pp'\iota} W_{ip't}^{pr} - \sum_{p \in P} \sum_{i \in N} Z_{it}^{pr} + n_t^{\iota r} \quad \forall \iota \in K, t \in T, r \in R \quad (6.39)$$

$$\sum_{i \in N_{p\iota}, i \neq j} (E_{jit}^{\iota r} - E_{ijt}^{\iota r}) = Z_{jt}^{pr} \quad \forall j \in N_{p\iota}, t \in T, r \in R, p \in P : \theta_{p\iota} = 1 \quad \iota \in K \quad (6.40)$$

$$\sum_{\substack{p' \in P \\ p' \neq p}} (E_{pp't}^{\iota r} - E_{p'pt}^{\iota r}) = \theta_{p\iota} R_{pt}^{\iota r} + \sum_{i \in N} \sum_{p' \in P} \epsilon_{pp'\iota} W_{ip't}^{pr} \quad (6.41)$$

$$\forall p \in P, \iota \in K, t \in T, r \in R$$

$$0 \leq \sum_{r \in R} \mathcal{L}_{it}^{Er} \leq c_\iota^E \quad \forall \iota \in K, t \in T \quad (6.42)$$

$$\sum_{p \in P} \sum_{r \in R} X_{ijt}^{p,r} \leq Q \sum_{p \in P} \sum_{v \in V} x_{ijvt}^p \quad \forall i, j \in N_p, i \neq j, t \in T \quad (6.43)$$

$$\sum_{r \in R} E_{ijt}^{\iota,r} \leq Q \sum_{v \in V} y_{ijvt}^\iota \quad \forall i, j \in N_{p\iota}, i \neq j, \iota \in K, t \in T \quad (6.44)$$

$$\sum_{r \in R} E_{pp't}^{\iota,r} \leq Q \sum_{v \in V} y_{pp'vs.t}^\iota \quad \forall t \in T, p, p' \in P, p \neq p', \iota \in K : \epsilon_{pp'\iota} = 1 \quad (6.45)$$

$$\sum_{i \in N_{p\iota}} \sum_{v \in V} y_{ijvt}^\iota \leq 1 \quad \forall j \in N, t \in T, \iota \in K \quad (6.46)$$

$$\sum_{i \in N_{p\iota}, i \neq j} y_{ijvt}^\iota = \sum_{i \in N_{p\iota}, i \neq j} x_{jivt}^\iota \quad \forall vs. \in V, j \in N_{p\iota}, t \in T \quad (6.47)$$

$$\sum_{j \in N} y_{0,jvt}^\iota \leq 1 \quad \forall v \in V, t \in T, \iota \in K \quad (6.48)$$

$$\sum_{p \in P_{p\iota} \cup 0_\iota} \sum_{v \in V} y_{pp'vs.t}^\iota \leq 1 \quad \forall p' \in P_{p\iota}, t \in T, \iota \in K \quad (6.49)$$

$$\sum_{p' \in P_{p\iota} \cup 0_\iota, p' \neq p} y_{p'pvt}^\iota = \sum_{p' \in P_{p\iota} \cup 0_\iota, p' \neq p} y_{pp'vt}^\iota \quad \forall vs. \in V, p \in P_{p\iota}, t \in T \quad (6.50)$$

$$\sum_{p \in P_{p\iota}} y_{0,pvt}^\iota \leq 1 \quad \forall v \in V, t \in T, \iota \in K \quad (6.51)$$

with:

$N_{p\iota}$: set of customers for whom the supplier p belongs to the cluster of centre ι

(node 0_ι).

$P_{p\iota}$: set of suppliers belonging to the cluster of centre ι .

$R_{pt}^{l,r}$: quantity of empty RTIs of type r belonging to supplier p and sent to centre ι to which supplier p belongs.

$E_{ijt}^{l,r}$: quantity of empty RTIs of type r transported from node i to node j in period t and sent to centre ι .

$E_{p'pt}^{l,r}$: quantity of empty RTIs of type r transported from node p to node p' in period t and sent by centre ι .

$\mathcal{L}_{it}^{E,r}$: inventory level of empty RTIs of type r at centre ι in period t .

y_{ijvt}^l : binary variable equal to 1 if node j is visited right after node i by vehicle v , 0 otherwise.

$y_{0_\iota jvt}^l$: binary variable equal to 1 if customer j is visited by v from node (cluster) 0_ι , 0 otherwise.

$y_{pp'vs,t}^l$: binary variable equal to 1 if supplier p' is visited right after supplier p by vehicle v , 0 otherwise.

$y_{0_\iota pvt}^l$: binary variable equal to 1 if supplier p is visited by v from node (cluster) 0_ι , 0 otherwise.

6.4 . Resolution Approach

The DM, SM, and IRPPDS models described in the previous section are NP-hard. To tackle their combinatorial complexity, a resolution approach is proposed.

We aim at determining over a given planning horizon the required quantities of RTIs to allow for supplying the needed quantities of products from a set of suppliers to a set of customers. We also seek to construct the optimal routes for pickups and deliveries of RTIs. Since the construction of the routes is the most complex part of the problem, we first use an appropriate heuristic to determine those routes. Once constructed, we solve a modified version of the three MILPs described in Section 3 to determine the other decision variables related, for example, to the quantities transported, delivered, and collected. Each of these versions is a min-cost network flow problem that is easier to solve. Regarding IRPPDS, its modified version is called FMILP, where the routing decision variables, x_{ijvt}^p , are fixed:

FMILP:

$$\begin{aligned} \min \sum_{i \in N} \sum_{t \in T} \sum_{r \in R} (h_i^{Lr} L_{it}^{Lr} + h_i^{Er} L_{it}^{Er}) + \sum_{p \in P} \sum_{t \in T} \sum_{r \in R} (H_p^{Lr} I_{pt}^{Lr} + H_p^{Er} I_{pt}^{Er}) + \\ \sum_{p \in P} \sum_{t \in T} \sum_{r \in R} e_r n_t^{p,r} + \sum_{p \in P} \sum_{t \in T} \sum_{p' \in P} \sum_{r \in R} g_r F_{pt}^{p'r} + \\ \sum_{i \in N} \sum_{p \in P} \sum_{p' \in P} \sum_{t \in T} \sum_{r \in R} s_r W_{ip't}^{pr} \end{aligned} \quad (6.52)$$

subject to Constraints (6.2)–(6.11).

We use a matheuristic to construct routes and improve the final solution as described above. The matheuristic hybridises the FMILP with an artificial-immune-system-based algorithm and a deep Q-learning process into a global solving scheme

called AIS-DQL. The overview of the matheuristic AIS-DQL is presented in Figure 6.1.

These steps are described in detail in the following subsections.

6.4.1 . Artificial Immune System

Artificial-immune-system-based algorithms are bio-inspired metaheuristics that imitate the principles and processes of immune system functioning (Talbi, 2009). The algorithms are typically modelled after the immune system's characteristics of learning and memory for use in problem-solving. They imitate antigen recognition, antigen and antibody binding, and antibody production. Furthermore, they abstractly use the diversity and memory mechanism of the immune system. Therefore, they can ensure individual diversity while maintaining a high affinity, avoiding premature phenomena and showing a strong global search ability. In this paper, antigens correspond to the input data of the problem, and the antibodies correspond to the routes to construct or the different suppliers. Their structure, depicted in Figure 6.2, consists of sequences of possible nodes to be visited in each route and for each supplier and each period. These steps are described in detail in the following subsections.

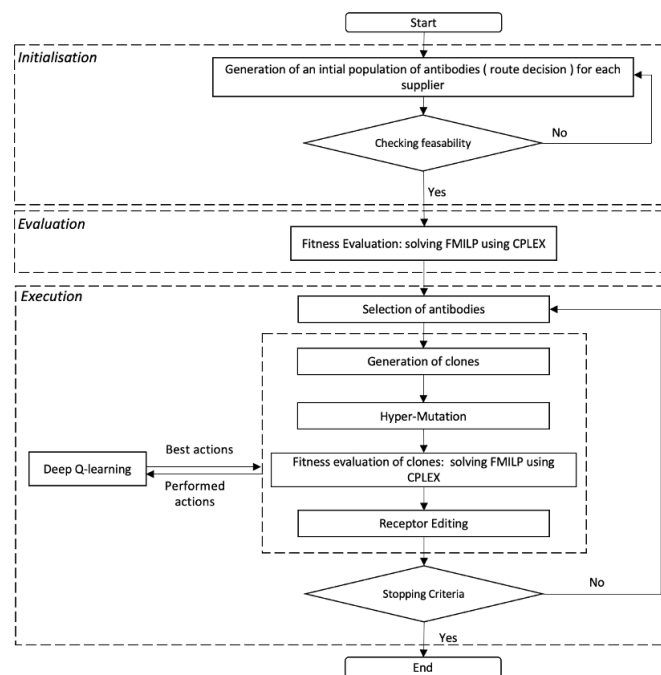


Figure 6.1: Overview of the implementation of the AIS-DQL matheuristic.

Supplier 1	Customer 1	Customer 2	----	Customer n-1	Customer n
Supplier 2	Customer 3	Customer 10	----	Customer n-10	Customer n-5
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
Supplier m	Customer n-2	Customer n	----	Customer 1	Customer 2

Figure 6.2: Structure representing the route of Supplier i (antibodies).

As depicted in Figure 6.1, AIS starts with an initialisation phase. A population of random routing solutions representing a pool of antibodies (routes) is initially generated. The routes are built using a 2-opt local search algorithm (Chiang et al., 2010). Proliferation and maturation processes are undergone by cloning each member of the initial pool, i.e., copying each of the initial solutions based on their affinity. The rate of proliferation is chosen to be directly proportional to the affinity, such that the higher the affinity, the more offspring there are. For this purpose, selection, Hyper-Mutation (HM), and receptor editing (RE) operators are used.

6.4.1.1 . Affinity and Cloning Selection

Each time an antibody (routing decisions) is generated, it is used as an input to solve the FMILP. Therefore, the corresponding feasible objective function (OF) and the remaining decision variables of the model are computed. The affinity f_ι of an antibody, ι , is computed using the corresponding objective function OF_ι : $f_\iota = \frac{1}{OF_\iota}$. Thus, the higher affinity value would have a lower total cost. Hence, as an antibody's cloning rate is proportional to its affinity, more antibody clones have lower costs in the next generation than antibodies with higher costs. The probability, PS, of selecting an antibody to be cloned depends on its affinity. If f_ι is the affinity of an antibody ι in the population, its probability PS_ι is defined as: $PS_\iota = \frac{f_\iota}{\sum_\varsigma f_\varsigma}$.

6.4.1.2 . Affinity Maturation

Since the algorithm needs to thoroughly explore and exploit the search space to obtain a good solution, exploration and exploitation depend on the evolution operator's capability variation. These operators conduct random perturbations on each gene to generate the next generation's population in the current population. The variation in the antibodies is performed through HM and RE mechanisms. The HM mechanism ensures that the higher-affinity antibodies are hyper-mutated at a slower rate. The HM_ι the rate for an antibody ι is defined as $HM_\iota = e^{-\omega f_\iota}$, where ω is the decay control factor. A new population is created after hyper-mutation, and each antibody undergoes various affinity changes. Antibodies are therefore reorganised once again based on the affinity assessment.

After cloning and mutation processes, a percentage of the antibodies in the current population is eliminated (the worst $\phi\%$ of the population) and replaced by the randomly generated antibodies. This mechanism, a vertebrate immune system mechanism, is called receptor editing (De Castro et al., 2000). This mechanism creates new antibodies that correspond to the new search area of the search space. Exploring new search areas may help the algorithm to escape from local optima. The new antibody population then becomes the next generation of antibodies.

Finally, if a generation's objective function value does not improve over the previous generation, convergence is assumed to be achieved, and it is possible to retrieve the best equivalent antibody as the best solution, and the algorithm stops.

6.4.2 . AIS enhanced with Deep Q-Learning

This section highlights AIS limitations and presents an RL technique used to overcome them.

6.4.2.1 . AIS and RL

According to (Bernardino and Barbosa, 2009b), although many results have proved the convergence of AISs to a global optimum, a Markov-chain analysis shows weak convergence of the AIS algorithms. Indeed, due to the single-point random mutation of the antibody, AIS converges slowly, meaning that a given antibody selects a gene bit and changes its value randomly to some of the other selectable values. Moreover, it cannot retain any locally excellent gene blocks in some low-affinity antibodies because of other poor gene blocks. As a result, the search speed is low. From this stems the idea of using RL to tackle this problem. Indeed, since random searching leads to slow evolution and weak AIS convergence efficiency, environmental feedback signals and the updated action policy of deep Q-learning are used to construct an algorithm with strong self-adaptability and goal-driven performance.

In this paper, RL is employed to assist in analysing data on moves and recombination that have been performed to construct solutions to the problem. The goal is to extract meaningful information from this data to direct and improve the AIS's search performance and speed. Indeed, just like a human being, the agent that symbolises the antibody (solution to the problem) learns on its own to acquire successful strategies that result in the most significant long-term rewards. RL is a paradigm of learning by trial and error based entirely on rewards or penalties. The agent constructs and learns its information directly from moves it makes using operators such as HM and RE. RL is used to assist AIS in determining the optimal actions to take in terms of the best moves for each operator.

6.4.2.2 . Q-Learning

Q-Learning is a self-adaptive RL off-policy method characterised by strong environmental feedback signals (Alom et al., 2019). The fundamental idea is to use the feedback signal to adjust an agent's action policy to make the best decision when interacting with the environment (i.e., antibody space). The agent (i.e., antibody) arrives in different states based on actions (i.e., AIS operators). Actions determine positive and negative rewards. The concept behind Q-learning

is to put the agent in a series of state-action combinations, observe the rewards, and then change the predictions of a table (called a Q-table) to those rewards until the best policy correctly predicts them. As a result, the "Q" stands for quality, which indicates how effective a particular action is in earning a possible reward.

6.4.2.3 . Deep Q-Learning

Q-learning is a relatively basic and effective algorithm. However, it may be time-consuming, as the amount of memory required to save and update the Q-table grows with the number of states, and the amount of time needed to investigate each state to construct the appropriate Q-table is impracticable. In this paper, these Q-values are estimated using neural networks known as deep Q-learning (DQ). Accordingly, the state is an input, and the output is the Q-value of all potential actions. Once the network is trained, selecting the right action means comparing each action's possible rewards and choosing the best one.

6.4.2.4 . Deep Q-Learning Architecture

DQ begins by estimating random Q-values to explore the environment, as shown in Figure 6.3. DQ enhances its Q-value estimations by employing the same dual-action paradigm, with a present action having a current Q-predicted value and a target action with a target Q-value. The direction of the predicted Q-target values varies since the network and its weights are equal; they remain unchanged but may fluctuate with each update. The Q-target values are stabilised by employing a second network that has not been trained. After a pre-determined number of iterations, called C-iteration, the learned weights from the Q-predicted network are copied to the Q-target network. The DQ design has two neural networks (Q-predicted and Q-target) and an experience replay agent, as shown in Figure 6.3. For data generation during Q-network training, the experience replay interacts with the environment. These data contain all of AIS's operators' moves, which are recorded as $\langle st, a, \mathcal{R}, st' \rangle$ tuples (see notation below Equation (6.53)). Then, a sample is picked randomly from these data, consisting of a mix of older and more recent samples. This batch of training data is used in the Q-predicted and Q-target networks. The Q-predicted network takes the current state and moves out of each sample, and for that move predicts the Q-value. The Q-predicted value, the Q-target value, and the observed data sample reward are used to compute the loss for the Q-network training (see Equation (6.53)). A batch of data is selected from all prior experiences to reduce variance and guarantee the algorithm's stability in C-iteration. Next, a sample is chosen from these data, with an equal mix of older and newer samples. The Q-predicted and Q-target networks use this batch of training data. The Q-predicted network predicts the Q-value based on the current state and moves out of each sample. The loss for the Q network training is computed using the Q-predicted value, the Q-target value, and the observed data sample reward (see Equation (6.53)). After each C-iteration a batch of data is selected from all prior experiences to reduce variation and ensure the algorithm's stability.

$$Loss = [\mathcal{R}_{t+1} + \gamma \max_a (\theta^T Q(st', a') - \theta^T Q(st, a))]^2 \quad (6.53)$$

where:

- γ : discount-rate parameter to measure the weight of the future awards.
- a, a' : current and future action, respectively.
- st, st' : current and future state, respectively.
- \mathcal{R}_{t+1} : future reward.
- $Q(st, a)$: learned action-value function.
- θ^T : transpose matrix of network weights.

Finally, as for the AIS memory, a set of the best antibodies having the highest affinity is stored, and the best moves have been obtained so far. Instead of starting from scratch every time the algorithm is run to solve the model for a given antigen, similar to the antigens (instances) already solved, we use the genetic memory to rapidly obtain the best solutions and the optimal policies for the antibodies. Similar antigens are selected based on the K-nearest neighbours algorithm (Mohtashami, 2015).

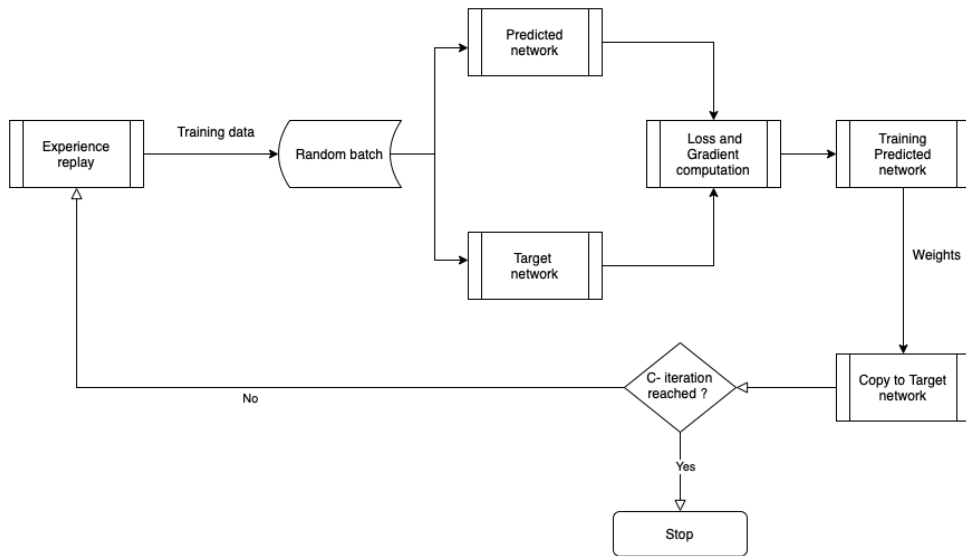


Figure 6.3: DQ architecture used as in (Achamrah et al., 2021)

6.5 . Implementation and Experimental Analysis

This section presents the experimental design adopted for this study and the analysis of the computational results. All the optimisation steps were carried out on a personal computer (MacBook Pro, macOS Cataline, CPU 3.3 GHz Quad-Core Intel Core i7, 8 GB of RAM).

Experimental Design and Parameters Tuning

The MILP developed for the multi-supplier, multi-customer, multi-RTI closed-loop supply chain was first solved optimally for small-and-medium-size instances using the Branch-and-Cut solver of CPLEX 12.9 (academic version). The objective was to check the model's validity, representativeness, and exact solving approach limitations.

To implement the metaheuristic, we used Python 3.7, and Pytorch interfaced with CPLEX. The approach was first tested on the same instances optimally solved using CPLEX. The objective was to assess its performance. We ran the AIS algorithm without the learning process (AIS) and compared the improvement provided by the deep Q-learning when coupled with the AIS algorithm (AIS-DQL). We also compared the algorithm's performance with a pure genetic algorithm (GA) and its improved learning version (GA-DQL). GA is also a population-based metaheuristic that mimics the principle of natural genetics to find a solution. The algorithm is known for its strong global search. The algorithm starts with an initial set of random solutions called a population. Each individual in the population is called a chromosome, representing a solution to the problem. The best parents (best chromosomes having the highest affinity) are selected from the current generation and considered for a two-point crossover operation to form their offspring. The mutation process is also integrated as it helps obtain new information randomly for the genetic search process and ultimately helps avoid getting trapped at local optima. In this paper, the chromosomes also represent the routing decisions and are decoded as the antibodies of AIS. For a thorough description of GA-DQL, the reader is referred to [Achamrah et al. \(2021\)](#).

The tests were performed in 20 replications for the 40 generated instances to evaluate the algorithms' stability, and the objective function's average value is presented. A statistical analysis using ANOVA was also conducted to assess the eventual randomness of the differences between the obtained results (see Table 6.3). These results stress that for all resolution approaches under consideration, p -value > 0.05 means no significant difference between the algorithms and the solutions obtained using CPLEX. Table 6.4 reports the algorithm parameters tuned so that a trade-off between the algorithm's performance and speed is satisfied.

Table 6.3: Statistical analysis using ANOVA.

Resolution Approach	F	$p - value$
GA	2.16	0.14
AIS	1.57	0.22
GA-DQL	0.91	0.34
AIS-DQL	0.49	0.49

Table 6.4: Values of the tuning parameters

Tuned Parameter	Value
Population size (GA/AIS)	200
Maximum iteration number (GA/AIS)	200
Crossover probability (GA/AIS)	0.81
Mutation probability (GA/AIS)	0.46
Selection probability	0.80
Receptor editing rate	0.28

The instances had a number of suppliers varying from 5 to 25, customers from 6 to 24, and RTIs ranging from 2 to 10. The planning horizon of deliveries and pickups was five days, corresponding to a workweek. Customer demands were randomly generated between 5 and 70 in loaded RTIs. For each instance, suppliers' and customers' locations were randomly chosen in the Euclidean space between (0,0) and (1000, 1000). Moreover, we considered initial inventory levels and unit costs for transportation, holding and maintenance of the self-same scale as in [lassinovskaia et al. \(2017\)](#), which considers a 1-M-1 IRP for a single type of RTIs. As the unit cost of an RTI may go from a few euros for plastic boxes to 1300 euros for stillages, according to the study conducted by [Limbourg et al. \(2016\)](#), we considered a randomly generated purchase cost varying between 3 and 1000 euros. Finally, we considered a unit cost of sharing ranging between 2 and 10 euros per type of RTI.

In the remainder of the paper, we refer to the instances using the following notation: (number of RTIs) R, (number of suppliers) S, (number of customers) C, (number of vehicles) V, and (number of periods) T; e.g., 1R2S5C2V5T refers to the instance where one type of RTI is shared and used to ship the products of two suppliers to a set of five customers, transported by two vehicles over a planning horizon of 5 days.

6.6 . Computational Experiments

First, the three models developed for the DM, SM, and IRPPDS modes, were solved using CPLEX. The objective was to compare the benefits and limitations of each mode on the performance of the overall supply chain. Then, given their combinatorial complexity, the three models were solved using four approaches: AIS and GA with and without DQL. The performance of each approach was analysed by comparing it to the solutions obtained using CPLEX on small instances. The benefits of DQL on the performance of the methods were also highlighted. Given the contribution of DQL, the three models were solved on large instances using the AIS-DQL and GA-DQL approaches.

6.6.1 . Results on Small Instances Solved Using CPLEX

The SM, DM, and IRPPDS models were first solved using the Branch-and-Cut solver of CPLEX until reaching optimality. We first considered solving the models with only one type of RTI for a number of customers varying from 6 to

24 in a planning horizon corresponding to a week of 5 days. We also conducted additional experiments in which we considered a number of RTIs varying from 2 to 10, and finally a number of suppliers varying between 5 and 25. The objective was to provide partial insights regarding the benefits of RTI sharing, the representativeness of the results, and the run time needed to solve this problem. Table 6.5 summarises the computational results for each instance under consideration. It reports the breakdown of the total cost (TC), namely: transportation (T), inventory of the suppliers (I-S), inventory of the customers (I-C), inventory at the centres for SM (I-K), maintenance (M), procurement of new RTIs (P), and sharing (S). Table 6.5 also provides the saving (%) between total costs for SM and IRPPDS regarding the total cost of DM and the CPU time in seconds. The saving, noted CS, is computed as follows: $CS = \frac{Total\ Cost_{DM} - Total\ Cost_{SM\ or\ IRPPDS}}{Total\ Cost_{DM}} 100$.

Table 6.5: Experimental results for DM, SM, and IRPPDS on small and medium instances solved using CPLEX.

Instances	Model	T (€)	I-S (€)	I-C (€)	I-K (€)	M (€)	P (€)	S (€)	TC (€)	CS (%)	CPU (s)
1R2S6P40V5T	DM	106,899	1386	1428	0	141	280,224	0	390,078	-	424
	SM	105,309	592	1368	751	133	122,771	2371	233,294	40	629
	IRPPDS	84,854	1188	1308	0	141	98,739	865	187,095	52	451
1R2S12P40V5T	DM	315,279	3706	1721	0	299	907,085	0	1,228,090	-	5050
	SM	245,866	2032	1359	2390	331	737,268	8940	998,186	19	6445
	IRPPDS	228,259	3562	1190	0	294	415,480	2998	651,784	47	5265
1R2S18P40V5T	DM	519,475	3067	4988	0	366	831,051	0	1,358,947	-	8776
	SM	471,330	1710	2680	1765	351	731,678	5504	1,215,018	11	12,115
	IRPPDS	402,425	2532	1911	0	352	611,920	4835	1,023,975	25	9331
1R2S24P40V5T	DM	853,012	4136	8040	0	685	3,280,781	0	4,146,653	-	24,314
	SM	711,300	3230	4061	2890	688	2,758,063	7761	3,487,994	16	31,701
	IRPPDS	552,893	4744	3046	0	696	1,886,929	6107	2,454,415	41	24,399
2R2S5P40V5T	DM	267,334	1013	4607	0	152	1,306,998	0	1,580,105	-	473
	SM	203,254	629	2957	746	171	1,188,760	3240	1,399,757	11	591
	IRPPDS	158,771	928	2011	0	154	781,631	1326	944,821	40	496
4R2S5P40V5T	DM	575,795	2055	11,269	0	502	1,719,310	0	2,308,932	-	1309
	SM	508,117	1177	5829	1432	426	1,493,784	6928	2,017,692	13	1626
	IRPPDS	413,915	1859	5141	0	457	1,056,450	5324	1,483,147	36	1316
6R2S5P40V5T	DM	984,677	3886	11,401	0	645	1,953,350	0	2,953,958	-	3013
	SM	601,250	1993	13,834	3007	578	1,608,985	12,681	2,242,328	24	4131
	IRPPDS	571,245	3372	6552	0	562	1,499,512	14,711	2,095,952	29	4405
8R2S5P40V5T	DM	1,196,050	3559	19,337	0	799	2,649,398	0	3,869,143	-	5423
	SM	1,027,848	2099	17,123	3738	734	2,270,434	221,00	3,344,076	14	7346
	IRPPDS	704,453	3307	11,693	0	765	1,726,959	15,446	2,462,623	36	5968
10R2S5P40V5T	DM	1,536,319	7376	22,048	0	1005	3,674,097	0	5,240,844	-	7981
	SM	1,450,099	3844	23,893	9146	1122	2,852,572	45,735	4,386,411	16	10,675
	IRPPDS	878,014	6777	11,199	0	967	2,263,598	18,631	3,179,185	39	8687
1R5S5P40V5T	DM	223,327	1698	3473	0	257	414,556	0	643,312	-	8084
	SM	213,934	947	3137	964	241	340,187	3415	562,825	13	13,686
	IRPPDS	157,568	1591	1754	0	233	245,555	1440	408,141	37	8969
1R10S5P60V5T	DM	470,266	3753	6052	0	544	1,040,464	0	1,521,078	-	22,526
	SM	465,720	1744	5879	1869	537	487,739	4204	967,693	36	33,882
	IRPPDS	383,377	2661	5856	0	505	209,126	2136	603,661	60	24,005
1R15S5P40V5T	DM	1,018,250	6620	12,547	0	798	882,493	0	1,920,708	-	32,387
	SM	995,463	4898	6364	4941	845	677,583	18,544	1,708,639	11	41,501
	IRPPDS	715,381	6145	4966	0	823	612,775	7922	1,348,012	30	34,055
1R20S5P40V5T	DM	1,595,794	9500	21,299	0	1310	1,929,223	0	3,557,125	-	55,543
	SM	1,419,879	6413	9549	7967	1139	1,724,055	25,988	3,194,989	10	67,511
	IRPPDS	893,618	10,265	7933	0	1181	754,922	9303	1,677,222	53	55,135
1R25S5P40V5T	DM	2,251,175	11,306	28,742	0	1488	1,758,976	0	4,051,687	-	67,115
	SM	2,044,658	6530	25,480	5387	1388	1,510,890	17,212	3,611,545	11	85,413
	IRPPDS	1,439,157	11,087	14,999	0	1536	1,394,438	13,932	2,875,148	29	63,468

From Table 6.5, we can see that, as expected, SM and IRPPDS reduce total costs compared to DM. Moreover, IRPPDS can help achieve significant cost savings; for IRPPDS, the average total cost was reduced by 40% against 17% for SM.

Indeed, in DM, each supplier needs to manage his inventory, deliveries, and the pickups of his empty RTIs from customers. As no shortage is permitted, if his inventory of empty RTIs is insufficient to meet customer demand, he buys this needed quantity, and a procurement cost is then incurred. Furthermore, regarding transportation costs, as each supplier can only use his RTIs, which cannot be shared among suppliers, the cost of picking up these latter from customers is incurred.

In SM, by contrast, as empty RTIs are owned and centrally managed by a pooler company, procurement costs could be reduced thanks to the risk pooling effect. Transportation costs (which include a variable cost that depends on the quantity of RTIs transported) are slightly reduced. Indeed, deliveries incurred by the suppliers remain the same as in DM, but not for the pickups of empty RTIs from customers. These empty RTIs are later transported to RTI centres owned by the pooler company, which are assumed to be located near suppliers, and they are transported to the suppliers when required. However, in SM, since the requests of RTI are not balanced between the suppliers, the pooler company must buy the needed quantities and ship them to the suppliers, which increases transportation and procurement costs.

As for IRPPDS, the transportation and procurement costs are significantly reduced. Indeed, in this configuration, the supply chain is centrally managed, and each supplier has his RTIs held at his inventory/customers and picks up empty ones from customers when vehicles visit customers to deliver the required products. In addition, each supplier can also benefit from this visit to pick up not only his RTIs but also the RTIs of other suppliers, and vehicle fill rates are improved (as transportation cost includes a variable cost that depends on the quantity of RTIs transported), and each supplier no longer needs to buy the RTIs he may need to meet his customers' demand. Orders are thus satisfied by any RTI, and a procurement order is only triggered if required, which reduces its relative cost compared to DM.

Moreover, in IRPPDS, the risk pooling is maintained as each supplier, when he buys RTIs, adds them to the system pool. Furthermore, the additional management costs of SM, including pooling, inventory, and transportation on the level of the pooler company's centres, are no longer incurred. Furthermore, as is shown in Figure 6.4, compared to IRPPDS, the quantity of new RTIs bought may represent, for some instances, up to 70% of the available inventory of empty RTIs in SM. More RTIs are purchased at each centre to meet the needs of the suppliers it serves. Finally, from Table 6.5 and Figure 6.4, we notice that IRPPDS takes on more interest as the number of RTIs and suppliers increases. Indeed, in SM, when the number of suppliers increases, more centres are needed, especially in different and distant geographical areas, making it challenging to reduce the cost of logistics and procurement for centres servicing clusters housing a significant number of suppliers. As a result, the demands for empty RTIs are not balanced; procurement, inventory, and transportation costs increase as the number of visits from customers to these centres and from these centres to suppliers increases.

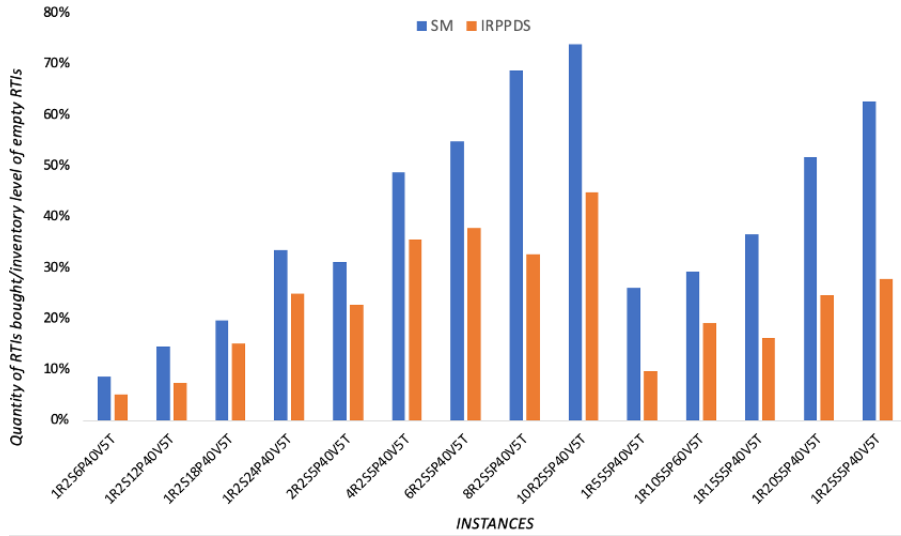


Figure 6.4: Ratio of quantity of RTIs bought over the inventory level of empty RTIs made available in SM and IRPPDS.

6.6.2 . First Insights into the Effectiveness of the Resolution Approach on Small Instances

As we can see, solving exactly the three models under consideration is very combinatorially complex, and the CPU time increases drastically with the number of suppliers. As described in the experimental design, the resolution approach AIS-DQL was compared to other metaheuristics to assess its performance: GA, AIS, and GA-DQL. Table 6.6 gives the results of the comparison. The gap regarding total cost is computed as: $Gap = \frac{Total\ Cost\ Metaheuristic - Total\ Cost\ CPLEX}{Total\ Cost\ CPLEX} 100$. From Table 6.6, we can see that for all the instances under consideration, AIS-DQL can find solutions with minor gaps compared to AIS, GA-DQL, and GA. On average, GA provided solutions with a gap of 12.6%, AIS with 9.4%, GA-DQL with 4.8%, and AIS-DQL with a gap of 0.1%. Indeed, AIS-DQL was more stable, as it was less sensitive to small changes (perturbations) in the input data and the instances' size. Moreover, AIS-DQL allowed for reducing the computational time considerably. GA and AIS may have similar mutation mechanisms, but AIS's immune memory makes it more robust and stable. Furthermore, AIS learning requires increasing the relative population size of each of these antibodies, which proved valuable. A clone is generated temporarily, and those low-affinity antibodies are eliminated. The goal is to solve the problem using minimal resources and time. Therefore, the algorithm's response efficiency was greatly enhanced by the memory associated with the first and best antibodies obtained for different and similar antigens. It was capable of providing the best solutions with a high affinity for a given instance only after a few iterations. Indeed, our algorithm ensures that both the speed and accuracy of the immune response are progressively higher after each model resolution. In addition, combined with a deep reinforcement learning technique and KNN, the immune memory further strengthens the interaction with

the environment, resulting in a continuous improvement of the algorithm's ability and prior knowledge of similar problems to solve the model for a given instance.

Table 6.6: Assessing the performance of GA, AIS, GA-DQL, and AIS-DQL compared to CPLEX on relatively small and medium instances.

Instance	Model	CPLEX			GA			AIS			GA-DQL			AIS-DQL		
		TC (€)	CPU (s)	Gap (%)	TC (€)	CPU (s)	Gap (%)	TC (€)	CPU (s)	Gap (%)	TC (€)	CPU (s)	Gap (%)	TC (€)	CPU (s)	Gap (%)
1R2S6P40V5T	DM	390,078	424	396,319	8	1.6	395,929	28	1.5	396,319	15	1.6	390,078	2	0.0	
	SM	233,294	629	257,557	9	10.4	254,757	22	9.2	245,892	10	5.4	234,461	31	0.5	
	IRPPDS	187,095	451	204,308	9	9.2	199,443	41	6.6	200,379	14	7.1	187,095	4	0.0	
1R2S12P40V5T	DM	1,228,090	5950	1,331,250	256	8.4	1,312,828	240	6.9	1,251,424	421	1.9	1,228,090	25	0.0	
	SM	998,186	6445	1,044,102	427	4.6	1,038,113	295	4.0	1,032,124	307	3.4	1,003,177	21	0.5	
	IRPPDS	651,784	5265	677,204	401	3.9	689,587	57	5.8	682,418	461	4.7	651,784	21	0.0	
1R2S18P40V5T	DM	1,358,947	8776	1,498,919	84	10.3	1,382,049	32	1.7	1,394,280	89	2.6	1,358,947	18	0.0	
	SM	1,215,018	12,115	1,364,466	527	12.3	1,362,036	62	12.1	1,273,339	57	4.8	1,219,879	77	0.4	
	IRPPDS	1,023,975	9331	1,135,888	824	10.9	1,087,461	925	6.2	1,054,694	483	3	1,023,975	50	0.0	
1R2S24P40V5T	DM	4,146,653	24,314	4,486,679	1394	8.2	4,470,092	884	7.8	4,457,652	1674	7.5	4,146,653	47	0.0	
	SM	3,487,994	31,701	3,857,721	989	10.6	3,864,697	508	10.8	3,808,889	416	9.2	3,508,922	21	0.6	
	IRPPDS	2,454,415	24,399	2,947,752	2313	20.1	2,923,208	25	19.1	2,648,314	455	7.9	2,454,415	163	0.0	
2R2S5P40V5T	DM	1,580,105	473	1,922,988	12	21.7	1,783,939	37	12.9	1,685,972	30	6.7	1,581,685	1	0.1	
	SM	1,399,757	591	1,542,532	40	10.2	1,511,738	12	8.0	1,426,353	20	1.9	1,402,557	42	0.2	
	IRPPDS	94,4821	496	1,037,413	18	9.8	993,952	4	5.2	963,717	35	2	945,766	3	0.1	
4R2S5P40V5T	DM	2,308,932	1399	2,613,743	75	15.8	2,542,134	76	10.1	2,403,598	4	4.1	2,308,932	12	0.0	
	SM	2,017,692	1626	2,350,700	39	17	2,209,373	41	9.5	2,031,816	51	0.7	2,017,692	22	0.0	
	IRPPDS	1,483,147	1316	1,739,731	124	17.3	1,576,685	21	6.3	1,566,203	79	5.6	1,483,147	5	0.0	
6R2S5P40V5T	DM	2,953,958	3013	3,202,090	63	8.4	3,190,275	42	8.0	3,140,057	227	6.3	2,953,958	6	0.0	
	SM	2,242,328	4131	2,679,583	58	19.5	2,684,067	6	19.7	2,419,472	8	7.9	2,249,055	53	0.3	
	IRPPDS	2,095,952	4405	2,290,876	70	9.3	2,292,971	379	9.4	2,179,790	240	4.0	2,098,048	2	0.1	
8R2S5P40V5T	DM	3,869,143	5423	4,604,280	538	19.0	4,016,170	527	3.8	3,927,180	392	1.5	3,873,012	29	0.1	
	SM	3,344,076	7346	3,691,860	525	10.4	3,410,957	604	2.0	3,390,893	209	1.4	3,347,420	41	0.1	
	IRPPDS	2,462,623	5968	2,856,643	350	16	2,570,978	46	4.4	2,561,128	31	4.0	2,462,623	44	0.0	
10R2S5P40V5T	DM	5,240,844	7981	5,801,614	733	10.7	5,848,782	670	11.6	5,382,347	293	2.7	5,246,085	28	0.1	
	SM	4,386,411	10,675	4,618,891	319	5.3	4,614,504	400	5.2	4,496,071	415	2.5	4,425,888	32	0.9	
	IRPPDS	3,179,185	8687	3,795,947	447	19.4	3,808,664	206	19.8	3,344,503	177	5.2	3,179,185	60	0.0	
1R5S5P40V5T	DM	643,312	8084	702,497	421	9.2	714,720	63	11.1	656,178	30	2.0	643,955	27	0.1	
	SM	562,825	13,686	665,259	276	18.2	613,479	54	9.0	602,786	38	7.1	567,328	6	0.8	
	IRPPDS	408,141	8969	454,261	726	11.3	422,425	702	3.5	436,711	100	7.0	408,549	7	0.1	
1R10S5P60V5T	DM	1,521,078	22,526	1,718,818	422	13.0	1,630,596	1732	7.2	1,522,599	337	0.1	1,522,599	168	0.1	
	SM	967,693	33,882	1,065,430	307	10.1	1,065,430	980	10.1	1,047,043	475	8.2	969,628	52	0.2	
	IRPPDS	603,661	24,005	719,564	285	19.2	705,680	508	16.9	644,106	295	6.7	604,265	160	0.1	
1R15S5P40V5T	DM	1,920,708	32,387	2,214,576	2272	15.3	2,212,656	1433	15.2	2,058,999	2871	7.2	1,920,708	48	0.0	
	SM	1,708,639	41,501	1,942,722	525	13.7	1,905,132	1245	11.5	1,802,614	395	5.5	1,717,182	22	0.5	
	IRPPDS	1,348,012	34,055	1,419,457	2947	5.3	1,443,721	1598	7.1	1,376,320	949	2.1	1,349,360	289	0.1	
1R20S5P40V5T	DM	3,557,125	55,543	4,197,408	1656	18.0	4,030,223	5005	13.3	3,841,695	148	8.0	3,557,125	103	0.0	
	SM	3,194,989	67,511	3,600,753	910	12.7	3,466,563	406	8.5	3,252,499	510	1.8	3,201,379	12	0.2	
	IRPPDS	1,677,222	55,135	1,893,584	902	12.9	1,893,584	5312	12.9	1,769,469	451	5.5	1,677,222	296	0.0	
1R25S5P40V5T	DM	4,051,687	67,115	4,590,561	3358	13.3	4,517,631	2022	11.5	4,221,858	1636	4.2	4,051,687	524	0.0	
	SM	3,611,545	85,413	4,174,946	3140	15.6	4,113,549	400	13.9	3,795,733	211	5.1	3,611,545	42	0.0	
	IRPPDS	2,875,148	63,468	3,346,672	4064	16.4	3,197,165	3750	11.2	2,955,652	70	2.8	2,875,148	385	0.0	

6.6.3 . Extra Experiments on Large Instances Solved Using GA-DQL and AIS-DQL

To obtain more insights into the effectiveness of IRPPDS and AIS-DQL, we further ran tests on large instances and compared the results to those obtained using GA-DQL. We solved the DM, SM, and IRPPDS. Then, we computed the total cost and the corresponding savings. We present the results obtained within a CPU time of less than half an hour. The computational results are summarised in Table 6.7. Table 6.7 also reports the difference (Diff) between the total costs computed using GA-DQL and AIS-DQL as follows:

$$Diff = \frac{Total\ Cost_{GA-DQL} - Total\ Cost_{AIS-DQL}}{Total\ Cost_{GA-DQL}} 100.$$

As expected, the AIS-DQL allows for feasible solutions to large-sized problems within a reasonable time. AIS-DQL allowed for better solutions with an average of 14% compared to GA-DQL and with less time, with an average CPU of 479 s for DM, 557 s for SM, and 556 s for IRPPDS against 743 s for DM, 644 s for SM, and 697 s for IRPPDS. As for the results, for all the instances under consideration, IRPPDS reduced total cost compared to DM, with an average saving of 35% (against 16% for SM). Moreover, the benefits of promoting virtual pooling were highlighted when the number of RTIs and suppliers increased. Furthermore, if the demands to be satisfied required the use of several types of RTIs, the benefits of SM were smaller compared to those of IRPDPS and even DM (according to

the results of Zhang et al. (2015)). This was truer when the number of suppliers increased. Indeed, even if SM can reduce the long-distance transportation of empty RTIs compared to DM, when empty RTIs are not balanced, SM's transportation cost increases since it includes the incurred costs from customers to the pooler's centres from these latter to the suppliers. In addition, even if the centres are located near suppliers (often in the automotive industry), it would be challenging to balance the quantity of empty RTIs between all suppliers. Therefore, SM may work in favour of or against any supplier regardless of location, the demands they should meet, or the centres' number.

Table 6.7: Experimental results for large instances obtained using GA-DQL and AIS-DQL.

Instances	AIS-DQL				GA-DQL				CS (%)	Diff (%)									
	DM TC (€)	CPU (s)	SM TC (€)	IRPPDS CPU (s)	DM TC (€)	CPU (s)	SM TC (€)	IRPPDS CPU (s)											
15R15S15C90V5T	1,055,305	330	979,362	274	806,191	217	7	24	1,217,822	527	1,074,615	318	902,934	276	12	26	13	9	11
15R20S15C120V5T	2,698,916	824	2,407,132	475	1,827,508	254	11	32	3,191,344	1207	2,939,561	550	2,077,807	316	9	35	15	18	12
15R30S15C150V5T	5,826,310	44	5,075,948	1070	3,877,637	980	13	33	6,659,472	70	5,079,089	1203	4,554,346	1259	24	32	13	0	15
15R40S15C200V5T	13,140,456	981	12,054,126	335	7,920,967	391	8	40	15,715,985	300	14,922,266	372	9,006,139	473	5	43	16	19	12
15R50S15C350V5T	27,832,333	513	19,565,584	395	17,764,072	329	29	36	31,013,110	829	27,865,271	463	20,673,888	410	10	33	11	30	14
15R60S15C450V5T	64,334,836	135	58,442,222	165	42,490,648	256	9	34	72,248,021	227	71,165,121	197	49,884,021	329	2	31	11	18	15
15R70S15C550V5T	143,862,932	97	134,278,723	156	80,904,270	156	7	44	172,347,793	151	134,553,700	169	97,004,220	198	22	44	17	0	17
15R80S15C650V5T	301,974,330	717	268,660,125	922	200,455,201	145	11	34	339,721,346	105	286,184,629	946	225,315,446	1454	16	34	11	6	11
15R100S15C750V5T	641,210,086	398	606,987,781	443	429,635,068	414	5	33	721,361,347	615	612,861,772	471	514,273,176	536	15	29	11	1	16
15R200S15C2000V5T	1,291,975,415	106	832,110,939	694	663,928,176	751	36	49	1,536,158,768	165	1,135,671,285	852	794,722,027	927	26	48	16	27	16
15R300S15C3000V5T	2,721,938,343	728	2,426,821,880	863	2,082,860,769	665	17	23	3,401,423,158	1146	2,913,447,999	870	2,442,391,404	827	19	21	12	10	15
15R400S15C6000V5T	5,791,634,472	268	4,014,243,737	268	3,126,883,840	237	31	46	6,718,516,388	409	4,560,776,575	489	3,592,795,532	292	26	47	14	19	13
15R600S15C8000V5T	11,431,199,715	384	8,504,611,093	357	7,582,649,005	178	26	34	13,603,127,661	624	9,546,143,904	391	8,507,732,184	217	30	37	16	11	11
31R200S34C4000V5T	4,428,883	740	3,726,512	845	3,445,812	971	16	22	5,053,356	1105	4,036,091	954	3,979,913	1182	20	21	12	7	13
31R400S34C9000V5T	10,976,497	391	7,776,378	569	7,089,187	839	30	35	13,008,734	475	9,160,248	690	8,607,744	1033	30	35	16	16	17
31R600S34C13000V5T	23,333,219	917	16,916,596	395	14,939,103	312	27	36	27,368,732	1388	25,748,719	424	16,731,795	392	6	39	15	34	11
31R800S34C25000V5T	53,970,919	595	43,722,965	622	28,413,814	603	19	47	62,605,222	890	57,142,014	714	31,993,955	745	9	49	14	23	11
31R1000S34C40000V5T	120,114,465	331	97,779,909	724	65,566,251	938	19	45	138,972,202	492	103,271,419	858	77,288,000	1188	26	44	14	5	15
31R1500S34C50000V5T	265,313,611	515	251,439,000	284	210,348,468	214	5	21	315,457,883	796	299,527,277	346	243,538,246	277	5	23	16	16	14
31R1500S34C60000V5T	572,377,195	294	506,511,145	531	410,627,455	495	12	28	681,701,239	456	569,930,183	608	473,453,456	614	16	31	16	11	13
31R2000S34C90000V5T	1,227,011,398	942	1,134,146,280	346	959,955,051	172	8	22	1,426,797,407	1481	1,312,205,027	306	1,136,586,780	221	8	20	14	14	16
31R3000S34C100000V5T	2,529,324,214	561	2,398,032,183	563	1,745,537,723	435	5	31	3,030,130,408	869	2,645,201,253	697	2,049,461,287	556	13	32	17	9	15
31R4000S34C180000V5T	5,169,582,108	319	4,520,814,515	483	3,311,432,130	511	13	36	5,789,931,961	516	5,245,182,139	556	3,751,852,603	640	9	35	11	14	12
31R5000S34C300000V5T	11,030,961,666	244	8,843,361,928	547	6,925,391,214	728	20	37	13,006,916,910	376	12,369,475,117	614	7,984,201,046	925	0	39	15	32	13
31R7000S34C450000V5T	23,471,937,843	1027	18,265,998,726	1128	14,568,743,068	1110	22	38	26,734,537,203	1598	23,081,746,819	1420	17,278,529,279	1412	14	35	12	21	16
31R7000S34C500000V5T	49,800,010,407	933	32,665,152,291	1090	26,308,338,166	1160	34	47	57,270,011,968	1439	56,207,283,103	1185	31,096,455,712	1430	2	46	13	42	15
Average	4,489,886,892	479	3,287,441,168	557	2,646,228,016	556	17	35	5,485,154,402	743	4,686,635,081	644	3,092,258,327	697	14	35	14	16	14

6.6.4 . Sensitivity Analysis on Unit Cost

Considering that the performances may depend on the different unit costs, a sensitivity analysis was conducted, and the results are given in this section. Without loss of generality, we ran tests on the instance 10R20S30C. For each test, we considered three scenarios. The first scenario represents the case where the unit cost of sharing is significantly lower than the smallest unit cost of procurement ($s_r = 5$). The second one corresponds to the case where the unit cost of sharing is equal to a given unit cost of procurement ($s_r = 700$). Furthermore, the third scenario represents the case where the unit cost of sharing is significantly higher than the greater unit cost of procurement ($s_r = 1800$). Figures 6.5 and 6.6 depict the variation in cost reduction (CR) for different values of cost parameters. It is worth noting that variable transportation and procurement costs were chosen to conduct the sensitivity analysis due to their significant contributions to the total costs.

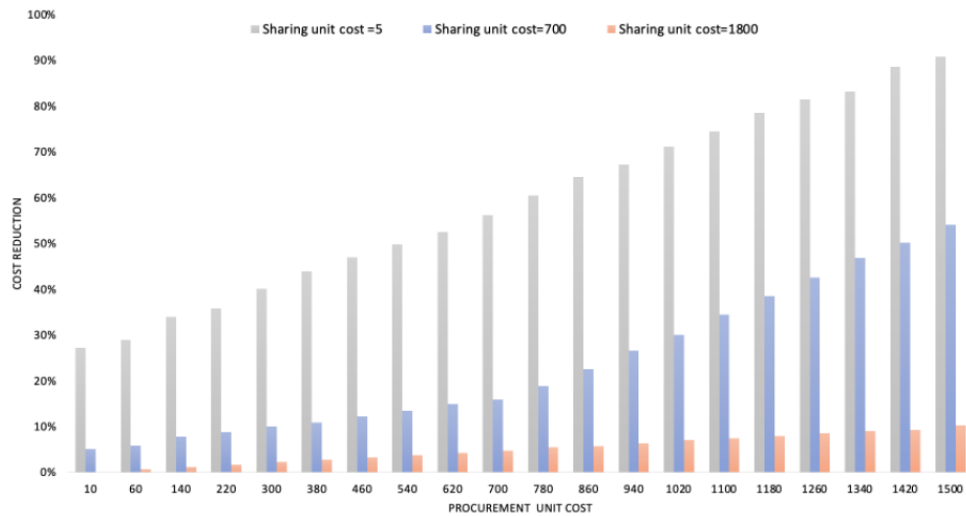


Figure 6.5: Saving in terms of total costs for the various unit costs of procurement.

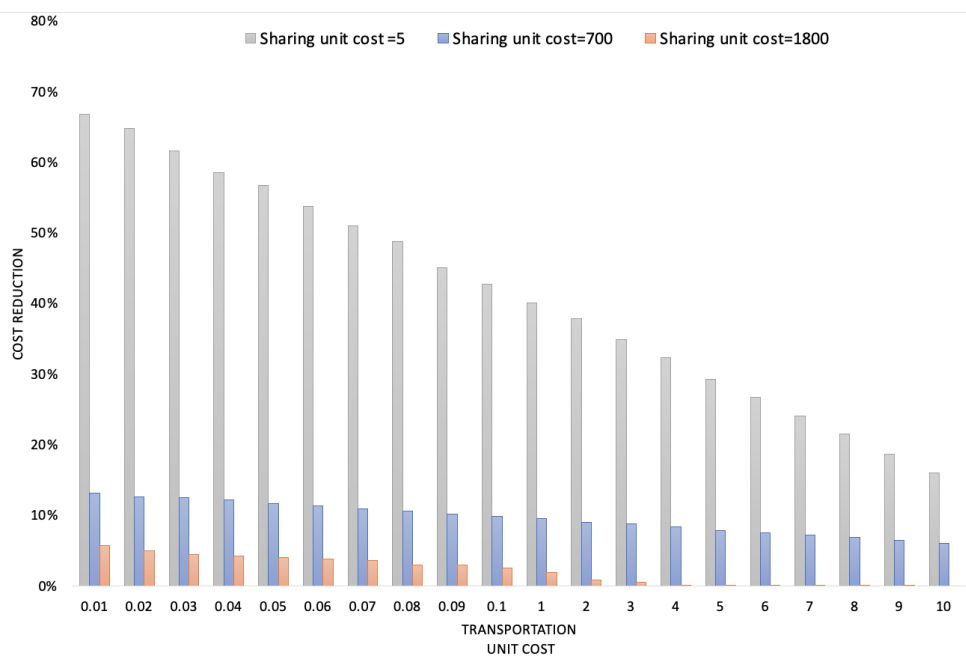


Figure 6.6: Saving in terms of total costs for the various unit costs of transportation.

From Figure 6.5, we see that from a cost-reducing perspective, IRPPDS generally has obvious advantages compared to DM for the three scenarios under consideration. Moreover, we notice that any change in the procurement cost has the most significant impact on cost reduction for IRPPDS. For the lower unit cost of

sharing, we see that as the procurement cost increases, the performance advantages of IRPPDS increase significantly. Indeed, authorising virtual pooling reduces inventory holding costs for empty RTI owners (lowering the idle stock of non-used empty RTIs), while suppliers who use these RTIs can meet more demands when no or fewer RTIs need to be bought. On the other hand, when the procurement cost is smaller than the unit cost of sharing, saving is smaller, and the advantages of DM and IRPPDS are comparable (this is even more evident in scenario 3). Thus, the benefits of IRPPDS may lessen as the cost incurred by sharing cannot be offset by the saving it brings regarding the reduction of procurement costs. Therefore, IRPPDS becomes profitable with a higher procurement cost. Moreover, as shown in Figure 6.6, significant savings are achieved when the sharing cost is smaller. In addition, when the transportation cost increases, IRPPDS is more profitable than DM. Indeed, more empty and loaded RTIs can be transported in a period (high fill rates), while fewer customers are visited, and fewer RTIs are bought in the next period. However, the predominance of IRPPDS may weaken when the sharing cost increases (this was even more evident in scenario 3, with GAP tending to zero). Indeed, with higher unit costs of sharing and transportation, it would be preferable and more cost-effective to have low fill rates (i.e., not to accept loading of unowned RTIs and to send them to the suppliers for further reuse) and to buy the needed RTIs rather than to have to pay for shared RTIs. Consequently, DM may be more profitable compared to IRPPDS.

6.7 . Conclusions and Perspectives

This paper considered a deterministic, multi-supplier, multi-customer and multi-RTI inventory routing problem with delivery and pickup in a collaborative supply chain in which empty RTIs inventories are virtually pooled among suppliers. We developed an MILP and solved it using CPLEX. Experiments showed that the virtual pooling of RTIs significantly reduces new RTI procurement costs and inventory and transportation costs compared to dedicated and shared modes. Moreover, to handle the combinatorial complexity of the problem, we developed an artificial immune-system-based algorithm coupled with deep reinforcement learning tailored to the mathematical program. We implemented our resolution approach using Python and Pytorch and compared it to the CPLEX solver and three metaheuristics: AIS without deep learning and GA with and without deep learning. Both variants of GA and AIS coupled with DQL seem to be competitive. However, the AIS variant outperformed GA thanks to its immune memory, which continuously improved the algorithm's speed and stability in solving the model. AIS-QDL even allowed for obtaining optimal solutions for some instances and feasible solutions with a tiny gap and within a small amount of time. Using AIS-QDL, we solved the model for large instances of up to 700 suppliers, 34 customers, and 31 types of RTIs. A sensitivity analysis of units' costs was also conducted. These results highlight how virtual pooling can be preferable compared to the dedicated and shared modes.

While the benefits of the model and the effectiveness of the AIS-DQL were

demonstrated using randomly generated instances, it would be beneficial to assess further their effectiveness on real data. Moreover, several possible applications may be investigated. For example, one could study the integration of cross docks in the RTI flows, as in the case of automotive supply chains. The idea is to combine and consolidate, when it seems advantageous, numerous smaller RTI loads provided by different suppliers and to deliver them downstream. Future research may also investigate the case of stochastic demands as room to exploit further and assess the limits of the resolution approach to tackle this kind of problem, the relative power of all parties in decision making, and maximising profit and its allocation. One way to address this latter may rely on the degree of commitment of the players. Indeed, as many supply chains experience the highest loss and damage rates of RTIs (which can be trackable using, for instance, RFID tags), the pool manager can reduce the costs incurred by "good" users and increase those of the "bad" ones or offer them training on the use of these RTIs so that they can improve on their weak points, reduce the environmental impacts, and increase the competitiveness of the whole system. Furthermore, decisions related to fleet composition and fuel consumption are to be considered in future work.

7 - Modeling and solving a stochastic multi-supplier multi-hub multi-customer IRP-T with shared inventory sharing of finished products and RTIs

Full paper to be submitted to International Journal Production Research, ISSN 00207543, 1366588X, Impact Factor: 8.6, Quartile: Q1.

Achamrah, F.E., Riane, F. and Aghezzaf, E.H. (2022). Integrating Forward and Reverse Network in the Interconnected Logistics Services in the Physical Internet.

Integrating Forward and Reverse Network in the Interconnected Logistics Services in the Physical Internet

Fatima Ezzahra Achamrah ^{a,b}, Fouad Riane ^{a,b}, El-Houssaine Aghezzaf ^{c, d}

^aComplex Systems and Interactions, Ecole Centrale of Casablanca; ^bLaboratoire Génie Industriel, CentraleSupélec, Paris Saclay University; ^c Ghent University, Faculty of Engineering and Architecture, Department of Industrial Systems Engineering and Product Design, ^d Industrial Systems Engineering (ISyE), Flanders Make

Abstract: We investigate the integration of forward and reverse networks in the interconnected logistics services in the Physical Internet (PI). A three-level PI-supply chain is considered, in which suppliers deliver their products to customers using reusable containers. In the light of sustainable development, forward and reverse flows are consolidated at the level of hubs. A mathematical model for the pickup and delivery of reusable PI-containers exploiting the PI concept is proposed. It considers inventory sharing between the PI-hubs, including empty and loaded PI-containers. Deliveries from suppliers to the PI-hubs and from PI-hubs to the customers are considered to be performed within time windows. A resolution approach based on the hybridization of mathematical modelling, artificial immune system algorithm and deep reinforcement learning is proposed to handle the combinatorial complexity of the problem at hand. Computational experiments highlight the positive impact of the PI model on logistics efficiency compared to the classical model, thanks to PI's high level of integration, flexibility and openness, which help achieve significant cost savings. They also stress that the resolution approach is very competitive and allows better solutions in terms of solution quality and run time.

Keywords: physical internet, reverse logistic, inventory routing with pickup and delivery, artificial immune system algorithm, deep reinforcement learning

7.1 . Introduction

The Physical Internet (PI) is a relatively new paradigm developed to enable the global logistics sustainability (Montreuil, 2011). Its vision focuses on 13 characteristics based on data transmission protocols that have formed the Internet (Montreuil, 2011; Ballot et al., 2014). PI is defined as how physical objects are transported, stored, supplied and used, aiming at greater efficiency and sustainability (Montreuil et al., 2012). PI, therefore, can introduce ground-breaking developments in material handling, logistics and facility design.

The main characteristic of PI is the use of globally standard, smart, green (either recyclable or reusable), shared and modular containers, also known as PI containers (Ballot et al., 2014; Montreuil, 2011; Montreuil et al., 2012). Ballot

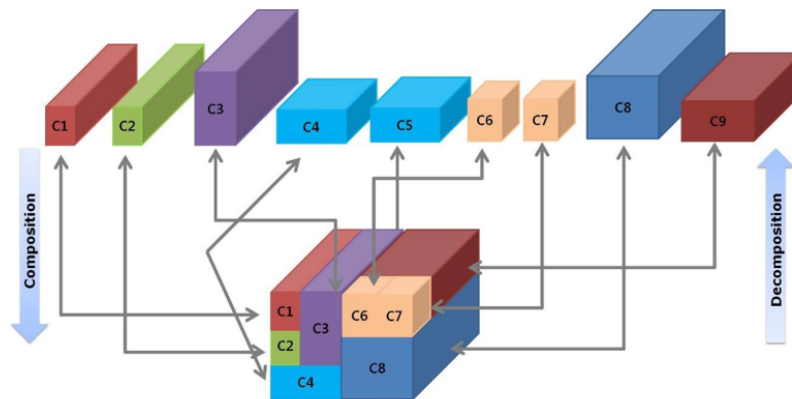


Figure 7.1: Modular PI-container for consolidation and de-consolidation (source: Montreuil (2011))

et al. (2014) discuss other detailed characteristics of the PI-containers. The authors emphasise that in the manner of the BIC code in the maritime sector, these containers have a unique international identification to ensure better traceability. PI-containers are also in a position to provide physical security and anonymisation of content which considerably facilitates the sharing of resources between various partners, including competitors. They also have a standardised size and mechanical strength, making it easier to handle and stack. And finally, by using a standardised system and appropriate twist-lock development, handling and locking between PI-containers are possible (see 4.2). PI containers can be therefore interlocked or encapsulated inside each other (Landschützer et al., 2015). These PI-containers will subsequently be decomposed into separate and smaller unit containers at PI-hubs for consolidation and better use of the capacity of trucks or better handling and storage at PI-hubs and destination locations (see Figure 7.1).

Aside from the PI-containers, movers and nodes are other critical physical elements of the PI (Fahim et al., 2021). Movers are grouped into PI-transporters, PI-conveyors and PI-sorters. For the nodes, PI-hubs and PI-transit centres are introduced. This forms PI's backbone. Therefore, the PI-logistics system can be defined as follows: if a shipment is made from supplier A to customer B, PI-containers are routed through one or more PI-hubs or PI-transit centres where they are consolidated, handled and transported by PI-conveyors, PI-sorters and PI-transporters. Similarly, when dealing with a set of suppliers and customers, goods with the same destination are consolidated and collected from and to transshipment points in the exact shipping period.

The PI logistics system is shared (Ballot et al., 2014) and thus, the expected benefit of such a system comes from sharing and pooling resources. Since the sharing of inventory and transportation resources allow for seamless interoperability, the cost, speed, quality and sustainability of the transport of goods are optimised (Tran-Dang et al., 2020). Routing systems are thus coordinated so that the means of transportation and their drivers meet at PI-hubs. The haulage and hauls can be

re-consolidated to avoid transports with empty or partially filled charge carriers and shorten transport times. Therefore, the PI's objective is to achieve higher utilisation of the transport routes. This is meant to provide significant economic (shorter transport times, fewer personnel costs) and ecological (traffic reduction, less CO₂ emissions) advantages (Montreuil, 2011). A balanced utilisation of storage facilities and distribution centres also strives (Ballot and Fontane, 2010; Pan et al., 2019). In such a system, routing and inventory decisions thus become very complex, and the need for optimisation models has become more than needed (Pan et al., 2019; Yang et al., 2017). This is even truer when dealing with reusable PI-containers, which involve the management of both forward and reverse flows. This paper aims to fill this gap.

This paper contributes to the literature in four dimensions: first, we develop an optimisation model for inventory routing of reusable PI-containers flowing in a closed loop. We investigate a PI-supply chain in which suppliers deliver their products using reusable PI-containers to their customers (e.g., plants, retailers). Direct and reverse flows of the PI-containers are consolidated at the level of a set of PI-hubs. Such management is highly relevant in the fast-moving consumer goods (FMCG) industry, where products are non-durable, delivered in packaged form, at low prices and in high volumes, and frequently purchased. Nevertheless, existing papers mainly consider traditional designs of the supply chain network, where each company establishes its logistics network independently (Iassinovskaia et al., 2017; Limbourg et al., 2016; Liu, 2020). This paper focuses on the inventory routing model under stochastic demand of reusable containers exploiting the PI concept, which is new to the literature. Second, we consider inventory sharing between the PI-hubs. Sharing includes both empty and loaded PI-containers. Third, we consider deliveries from suppliers to the PI-hubs and from PI-hubs to the customers to be performed within specified time windows as is usually done in practice. And finally, a resolution approach based on the hybridisation of mathematical modelling and an artificial immune system-based algorithm and reinforcement learning is proposed to handle the combinatorial complexity of the problem at hand. Also, thanks to its generality and flexibility, this matheuristic uses deep reinforcement learning techniques originally presented by Achamrah et al. (2021) to solve dynamic and stochastic inventory routing problems successfully.

In this paper, we aim to determine for each level of a three-level PI-supply chain, under stochastic demands and over a given planning horizon, the required quantities of empty and loaded PI-containers to allow the flows of products necessary to meet the customers' demand within specified time windows. The objective is to construct thus planning for PI-containers deliveries, pickups, transshipment and procurement of new ones while performing deliveries the predetermined within time windows.

The remainder of this work is structured as follows. In Section 2, related works are presented. Problem description and formulation are provided in Sections 3 and 4. Section 5 describes the resolution approach, while Section 6 reports the experimental results. Finally, conclusions and perspectives are presented in Section 7.

7.2 . Related work

Two research streams which are most aligned with our work address classical cross-docking into a supply chain environment and PI application in supply chain management. Regarding the first research stream, cross-docking takes place to consolidate shipments from disparate and even decentralised sources to achieve full-load outbound trucks and satisfy the customers' demands in a minimum delivery time. Many papers discuss Vehicle Routing Problem (VRP) with the classical cross-docking problem, and associated topics cover both models and solution approaches to deal with their combinatorial complexity. [Lee et al. \(2006\)](#) study a VRP with a cross-dock centre and propose a heuristic based on a tabu search algorithm to tackle the problem. [Benjamin \(2020\)](#) proposes a model to address a VRP with multi-product cross-docking and uses CPLEX to solve the model for small instances. [Wang et al. \(2017\)](#) propose a model to address VRP with multiple cross-docks and split deliveries. A constructive heuristic with two-layer simulated annealing and tabu search is used to solve the problem. [Musa et al. \(2010\)](#) address a VRP where products are shipped to customers directly or via cross-docks. They implement an ant colony optimisation algorithm to solve the problem at hand. [Mousavi and Vahdani \(2017\)](#) develop a two-phase deterministic model for VRP with multiple cross-docks and use a metaheuristic called a self-adaptive imperialist competitive algorithm to tackle the problem of concern. [Babae Tirkolaee Erfan Goli \(2020\)](#) propose a bi-objective model for a VRP with cross-docking. The model takes into account pollution and routing costs, and supply reliability. They use the multi-objective simulated-annealing algorithm and non-dominated sorting genetic algorithm to provide Pareto solutions to the problem. Other papers also integrate time windows constraints or transshipment between cross-docks. [Ma et al. \(2011\)](#) model a transportation problem with cross-docking and time windows and propose a heuristic approach to solve it. [Lim et al. \(2005\)](#) consider transshipment between cross-docks constrained by transportation schedules and warehouse capacities. [Chen et al. \(2006\)](#); [Marjani et al. \(2012\)](#) study the VRP with cross-docking and consider both transshipment and time windows constraints. Both papers use simulated annealing and tabu search based algorithms to solve the problem. We also note scarce interest in considering both forward and reverse flows of products. [Zuluaga et al. \(2016\)](#) analyse and adjust a set of models used to optimise forward flows to cope with reverse flows' characteristics in cross-docking. [Kheirkhah and Rezaei \(2015\)](#) develop a mathematical model that integrates returned products' flows. [Rezaei and Kheirkhah \(2017\)](#) considers an integrated supply chain network in which the forward/reverse cross-docking is applied, wherein the demands of customers (in forwarding flow) and that of recovery centres (in reverse flow) can be less than a truckload. The authors use the general algebraic modelling system (GAMS) software to solve the model. [Kaboudani et al. \(2020\)](#) study a VRP with a cross-docking centre and consider forward and reverse flows. A simulated annealing algorithm is used to solve the problem of concern.

On the other hand, previous papers related to PI have dealt chiefly with functional design for a road-based transit centre ([Meller et al., 2012](#)), tracking technol-

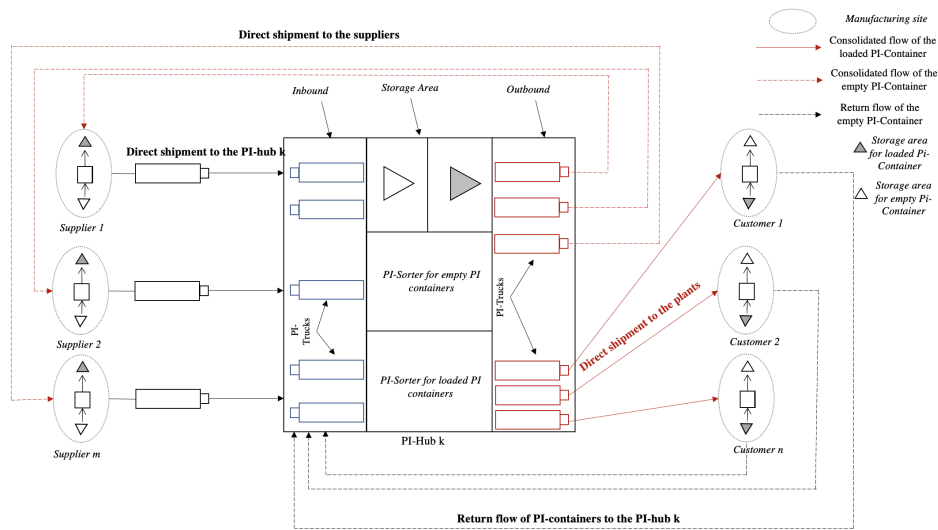


Figure 7.2: Forward and reverse flows of PI-containers in a PI-closed loop comprising set of suppliers, cross-docks, and customers

ogy for the management of returnable PI-containers (Roch et al., 2014), routing problem (Fazili et al., 2017), rule-based simulation models for inventory control (Pan et al., 2014), a single product inventory optimisation control model with transshipment (Yang et al., 2017) and PI-enabled integrated production inventory and distribution (Ji et al., 2019). It is also necessary, however, to consider cross-docking hubs and PI-containers management from an operational and sustainable viewpoint. The primary objective is to find the optimal inventory management, forward and reverse flows of reusable PI-containers and transshipment between cross docks, all constrained by time windows. To the best of our knowledge, no work has been reported that addresses such a PI-setting. Thus, this paper aims to fill this gap.

7.3 . Problem description

As a part of a PI-enabled vendor-managed inventory (VMI) system, the objective is to optimise transportation, inventory, transshipment and production costs for a uni-modal freight for delivery and pickup in a PI-closed loop supply chain. This supply chain consists of multiple suppliers and multiple customers and in which a set of PI-hubs are used to consolidate the forward and reverse flows of reusable PI-containers. This paper focuses on the cross-docking terminals, mainly Road–Road PI-hubs used to transfer the PI-containers between inbound and outbound PI-trucks (see Figure 7.2).

The planning horizon length is d with discrete-time periods $t \in T = \{1, \dots, d\}$. Regarding the sequence of the operations for forwarding flow, we assume that first, the decisions related to the quantities to be delivered by each supplier to PI-hubs are determined. Second, after customers reveal demands, they are supplied by PI-

hubs. Moreover, inventory sharing is allowed between PI-hubs to avoid, as much as possible, shortages at the level of each customer.

This paper assumes that direct shipping is not allowed from suppliers to customers and vice versa. All flows must be handled and consolidated at the PI-hubs before being shipped to their final destination (supplier or customer). Moreover, under VMI and Maximum Level policies (Yang et al., 2017), we assume that each supplier j can freely decide on the quantity to supply to the PI-hubs in terms of loaded PI-container p , restricted only by the customer i 's inventory capacity and by a threshold (reorder point) r_{pijt} . This quantity defines the parameter Q_{pijt} which is assumed to be proportional to $\max[0, r_{pijt} - I_{pit}^{L,0}]$, where $I_{pit}^{L,0}$ is the inventory level at a customer i at the beginning of a period t expressed in terms of loaded PI-container p .

As in Coelho et al. (2014a); Achamrah et al. (2021), we assume a normally distributed demand; a reorder point r can be then computed as follows:

$$r = \mu + \sigma g_{\beta} \quad (7.1)$$

Where μ at a given period is the expected demand and σ the related standard deviation, β is the shortage probability, and g_{β} is the β -order quantile of the demand distribution. We note that these values and r are updated in each period t . Also, the Expression 7.1 is applied before solving the model for each supplier, customer and loaded PI-container.

Transshipment between PI-hubs is permitted so that they can patch the demands by other's inventories. Transshipment can take place as an emergency measure to avoid shortages as much as possible whenever demands that have been revealed exceed the quantity of loaded PI-containers made available to each customer. Transshipment also includes sharing empty PI-containers to avoid packaging unavailability and unnecessary over-investment in buying new PI-containers. This transshipment is allowed whenever it seems profitable, and each supplier's required quantity of empty PI-containers exceeds his inventory level (insufficient to meet future demands).

Furthermore, we assume a direct shipment to customers from PI-hubs. After the demand is revealed, each customer is visited to be delivered the required quantity of loaded PI-containers. We also benefit from this visit to collect empty PI-containers necessary for future deliveries. Once returned, the empty PI-containers are sorted and consolidated at the level of PI-hubs, before being directly shipped to the suppliers for further reuse. Deliveries from suppliers to the PI-hubs and from PI-hubs to the customers are performed within time windows. We also assume that each location (supplier, customer and PI-hub) has two storage areas. One is dedicated to the storage of loaded PI-containers and the other for empty PI-containers. Each stock area is characterised by an initial inventory level and a maximum storage capacity. At the beginning of the planning horizon, each customer places an order in terms of loaded PI-containers to a PI-hub. The supplier has to directly ship the products to the given PI-hub within a time window. After being unloaded from inbound PI-trucks, the PI-containers are decomposed and sorted using PI-sorters. PI-containers having the same destination are then grouped and loaded into the

outbound PI-trucks (or after being stored temporarily in the storage zone). The outbound PI-trucks ship the PI-containers to the downstream customer and pick up the empty PI-containers to be returned to the PI-hubs for consolidation before sending them back to the suppliers for further reuse.

The transportation cost is computed based on flows. This paper considers five classes given the two flows of PI-containers: from suppliers to PI-hubs, from PI-hubs to suppliers, to customers, from customers to PI-hubs, and between PI-hubs.

Finally, aside from the assumptions already expressed, we consider the following:

1. Shortage penalty is incurred if demands are not met even after allowing lateral transshipment (Coelho et al., 2014a).
2. The time window restriction should be satisfied.
3. The time horizon is finite and is subdivided into equal periods.
4. The transshipment of PI-containers, either loaded or empty, between PI-hubs is possible all the time.
5. PI-trucks can load either empty or loaded PI-containers.
6. Buying new PI-containers is allowed if the quantity of empty PI-Containers remains insufficient to meet the needs of suppliers.
7. At the end of each period, all empty PI-containers are subject to quality control. Damaged ones are disposed of, serviceable PI-containers are repaired, and undamaged ones are transferred to the inventory.

The problem considers delivery and pickup, inventory and transshipment problems of PI-containers in a PI-closed loop supply chain. The objective is to determine for each level of the supply chain and over the planning horizon the quantities of PI-containers to be delivered by the suppliers and the quantities of empty PI-containers to be picked up or bough, if necessary. Deliveries to customers must be performed within time windows. To do this, we need to set up planning for the inventory level constraints (no shortages and overstocking are allowed) while taking into account time windows constraints, the availability of empty PI-containers at suppliers and PI-hubs, and minimising the inventory holding cost, operating cost at the PI-hubs, transportation, production/cleaning and new PI-containers purchasing costs.

7.4 . Problem formulation

Using the notations, parameters and decisions variables reported in Table 7.1, the pickup and delivery of PI-containers in a PI-hubs network with forward and reverse flows is formulated as follows:

$$\begin{aligned} \min \quad & \sum_{t \in T} \sum_{k \in K} \sum_{p \in P} (h_{pk}^L I_{pkt}^L + h_{pk}^E I_{pkt}^E) + \sum_{t \in T} \sum_{i \in N} \sum_{p \in P} (h_{pi}^L I_{pit}^L + h_{pi}^E I_{pit}^E) + \\ & \sum_{t \in T} \sum_{j \in S} \sum_{p \in P} (h_{pj}^L I_{pjt}^L + h_{pj}^E I_{pjt}^E) + \sum_{t \in T} \sum_{j \in S} \sum_{k \in K} c_{jk} x_{jkt}^L + \sum_{t \in T} \sum_{i \in N} \sum_{k \in K} c_{ki} y_{kit}^L + \end{aligned}$$

Table 7.1: The model's notation

Sets	
S	Set of suppliers
K	Set of PI-hubs
N	Set of customers
P	Set of PI-containers
T	Set of periods
Parameters	
c_{jk}	Transportation cost from a supplier j to a PI-hub k
c_{ki}	Transportation cost from a PI-hub k to a customer i
$c_{kk'}$	Transshipment cost from a PI-hub k to a PI-hub k'
h_{pk}^L, h_{pk}^E	Holding cost of a loaded and an empty PI-container p at the PI-hub k resp.
h_{pi}^L, h_{pi}^E	Holding cost of a loaded and an empty PI-container p at the customer i resp.
h_{pj}^L, h_{pj}^E	Holding cost of a loaded and an empty PI-container p at the supplier j resp.
b_p	Unit cost of buying a new PI-container p
g_p	Unit cost of maintaining and cleaning a PI-container p
a_{pi}	Unit cost of shortage per unit of loaded PI-container p at the level of customer i
TS_j^D, TE_j^D	Starting and ending time of delivering PI-containers from a supplier j resp.
TS_i^P, TS_i^E	Starting and ending time of shipping PI-containers to a customer i resp.
I_{pk0}^L, I_{pk0}^E	Initial inventory of a loaded and an empty PI-container p at a PI-hub k resp.
I_{pj0}^L, I_{pj0}^E	Initial inventory of a loaded and an empty PI-container p at a supplier j resp.
I_{pi0}^L, I_{pi0}^E	Initial inventory of a loaded and an empty PI-container p at a customer i resp.
V_p^L, V_p^E	Volume occupied by a loaded and an empty PI-container p resp.
C_k^L, C_k^E	Maximum holding capacity of PI-hub k (loaded and empty PI-containers resp.)
C_i^L, C_i^E	Maximum holding capacity of customer i (loaded and empty PI-containers resp.)
C_j^L, C_j^E	Maximum holding capacity of supplier j (loaded and empty PI-containers resp.)
D_{pit}	Customer demand which is normally distributed with average parameter μ_{pit} and standard deviation σ_{pit}
(r_{pijt}, Q_{pijt})	Replenishment policy of supplier j ; Q_{pijt} for batch size and r_{pijt} for reorder point considered by supplier j for each customer i and PI-container p
M	Big number
Variables	
x_{jkt}^L	Binary variable equal to 1 if supplier j is assigned for delivering loaded PI-containers to PI-hub k at period t , otherwise.
y_{ikt}^L	Binary variable equal to 1 if a PI-hub k is assigned for delivering PI-containers to customer i at period t , otherwise.
x_{jkt}^E	Binary variable equal to 1 if supplier j is assigned to PI-hub k for receiving empty PI-containers at period t , otherwise.
y_{ikt}^E	Binary variable equal to 1 if a PI-hub k is assigned to customer i for receiving empty PI-containers at period t , otherwise.
$z_{kk't}$	Binary variable equal to 1 if there is a transshipment between a PI-hub k and PI-hub k' at period t , otherwise
I_{pkt}^L, I_{pkt}^E	Inventory level of loaded and empty PI-container p at PI-hub k at period t resp.
$I_{pj't}^L, I_{pj't}^E$	Inventory level of loaded and empty PI-container p at supplier j at period t resp.
I_{pit}^L, I_{pit}^E	Inventory level of loaded and empty PI-container p at customer i at period t resp.
$w_{pkk't}^L$	Quantity of loaded and empty PI-container p shipped from PI-hub k to PI-hub k' at period t resp.
$w_{pkk't}^E$	
Q_{pkit}	Quantity of loaded PI-container p filled by the product of supplier j shipped from a PI-hub k to a customer i at period t
R_{pkijt}	Quantity of empty PI-container p shipped from a customer i to PI-hub k at period t
R_{pkjt}	Quantity of PI-container p to be shipped from a PI-hub k to supplier j at period t
n_{pkt}	Quantity of PI-container p being bought and stored in the PI-hub k at period t
$F_{pj't}$	Quantity of empty PI-container p being filled by products at the supplier j at period t
u_{pit}	Quantity of lost sales at customer i expressed in terms of the number of filled PI-container p at period t

$$\begin{aligned}
& \sum_{t \in T} \sum_{j \in S} \sum_{k \in K} c_{kj} x_{kjt}^E + \sum_{t \in T} \sum_{i \in N} \sum_{k \in K} c_{ik} y_{ikt}^E + \sum_{t \in T} \sum_{k \in K} \sum_{k' \neq k \in K} c_{kk'} z_{kk't} + \\
& \sum_{t \in T} \sum_{k \in K} \sum_{p \in P} b_p n_{pkt} + \sum_{t \in T} \sum_{j \in S} \sum_{p \in P} g_p F_{pj t} + \sum_{t \in T} \sum_{i \in N} \sum_{p \in P} a_{pi} u_{pit} \quad (7.2)
\end{aligned}$$

Subject to:

$$\begin{aligned}
I_{pkt}^L &= I_{pkt-1}^L + \sum_{i \in N} \sum_{j \in S} Q_{pjit} x_{jkt}^L - \sum_{i \in N} Q_{pkit} - \sum_{k' \neq k \in K} w_{pkk't}^L + \sum_{k' \neq k \in K} w_{pk'kt}^L \\
&\quad \forall p \in P, k \in K, t \geq 1 \quad (7.3)
\end{aligned}$$

$$\begin{aligned}
I_{pkt}^E &= I_{pkt-1}^E + \sum_{i \in N} R_{pkit} - \sum_{j \in S} R_{pkjt} - \sum_{k' \neq k \in K} w_{pkk't}^E + \sum_{k' \neq k \in K} w_{pk'kt}^E + n_{pkt} \\
&\quad \forall p \in P, k \in K, t \geq 1 \quad (7.4)
\end{aligned}$$

$$I_{pj t}^L = I_{pj t-1}^L - \sum_{i \in N} Q_{pjit} + F_{pj t} \quad \forall p \in P, j \in S, t \geq 1 \quad (7.5)$$

$$I_{pj t}^E = I_{pj t-1}^E + \sum_{k \in K} R_{pkjt} - F_{pj t} \quad \forall p \in P, j \in S, t \geq 1 \quad (7.6)$$

$$I_{pit}^L = I_{pit-1}^L + \sum_{k \in K} Q_{pkit} - \sum_{j \in S} D_{pj t} + u_{pit} \quad \forall p \in P, i \in N, t \geq 1 \quad (7.7)$$

$$I_{pit}^E = I_{pit-1}^E + D_{pit} - \sum_{k \in K} R_{pkit} \quad \forall p \in P, i \in N, t \geq 1 \quad (7.8)$$

$$0 \leq \sum_{p \in P} V_p^L I_{pkt}^L \leq C_k^L \quad \forall k \in K, t \in T \quad (7.9)$$

$$0 \leq \sum_{p \in P} V_p^E I_{pkt}^E \leq C_k^E \quad \forall k \in K, t \in T \quad (7.10)$$

$$0 \leq \sum_{p \in P} V_p^L I_{pit}^L \leq C_i^L \quad \forall i \in N, t \in T \quad (7.11)$$

$$0 \leq \sum_{p \in P} V_p^E I_{pit}^E \leq C_i^E \quad \forall i \in N, t \in T \quad (7.12)$$

$$0 \leq \sum_{p \in P} V_p^L I_{pj t}^L \leq C_j^L \quad \forall j \in S, t \in T \quad (7.13)$$

$$0 \leq \sum_{p \in P} V_p^E I_{pj t}^E \leq C_j^E \quad \forall j \in S, t \in T \quad (7.14)$$

$$\sum_{p \in P} \sum_{i \in N} Q_{pjit} \leq M \sum_{k \in K} x_{jkt}^L \quad \forall j \in S, k \in K, t \in T \quad (7.15)$$

$$\sum_{p \in P} Q_{pkit} \leq M y_{ikt}^L \quad \forall j \in S, k \in K, t \in T \quad (7.16)$$

$$\sum_{p \in P} (w_{pkk't}^L + w_{pkk't}^E) \leq M z_{kk't} \quad \forall k, k' \in K, k \neq k' \quad (7.17)$$

$$\sum_{p \in P} R_{pkjt} \leq M x_{jkt}^E \quad \forall j \in S, k \in K, t \in T \quad (7.18)$$

$$\sum_{p \in P} R_{pkit} \leq M y_{ikt}^E \quad \forall i \in N, k \in K, t \in T \quad (7.19)$$

$$\sum_{k \in K} \sum_{t \in TS_j^D}^{TE_j^D} x_{jkt}^L = 1 \quad \forall j \in S \quad (7.20)$$

$$\sum_{k \in K} \sum_{t \in TS_i^D}^{TE_i^D} y_{ikt}^L = 1 \quad \forall i \in N \quad (7.21)$$

The objective function (7.2) minimises the total cost. The three first sums correspond to the inventory holding cost at PI-hubs, suppliers and customers, respectively, for empty and loaded PI-containers. The fourth sum to the eighth sum corresponds to transportation costs from suppliers to PI-hubs, from PI-hubs to customers and between PI-hubs for both forward and reverse flows of empty and loaded PI-containers. Constraints (7.3)-(7.8) define at each location the conservation conditions for the conservation of inventory levels of loaded and empty PI-containers. Constraints (7.9)-(7.14) indicate the boundaries of the inventory levels of loaded and empty containers at the level of PI-hubs, suppliers and customers. Constraints (7.15) guarantee that the quantity of loaded PI-containers shipped from a supplier to a PI-hubs is positive only when the related binary variable is. Similarly, constraints (7.16)-(7.19) ensure the same logic between PI-hubs and between PI-hubs to customers for loaded and empty PI-containers. Constraints (7.20)-(7.21) stipulate that each delivery from suppliers to PI-hubs and from PI-hubs to customers is performed only within its time window.

7.5 . Resolution approach

The classical cross-docking problem is NP-complete in the strong sense (Chen et al., 2006). It is even harder for our setting, which integrates forward and reverse flows, transshipment and time windows constraints. Therefore, exact methods can fail to find optimal solutions for large-size problems. Given the complexity of the cross-docking problem, our approach is, therefore, to use a metaheuristic that combines mathematical modelling and an artificial immune system (AIS). For each flow, we determine using AIS the values of three decisions: the choice of the PI-hubs assigned for delivering full PI-containers and for receiving empty ones, and the time of delivery to the customers. The rest of variables are determined by solving the MILP in which the related variables of the three decisions are fixed. We note this MILP: FMILP

7.5.1 . Artificial immune system

To be used in problem-solving, AIS is modeled after the immune system's learning, and memory features (De Castro et al., 2000). Antigen recognition, antigen and antibody binding, and antibody synthesis are all replicated. It also uses the immune system's variety and memory mechanisms abstractly. As a result, it can assure individual variation while retaining a high level of affinity, preventing prema-

ture occurrences that demonstrate a great global search ability (De Castro et al., 2000). Finally, even though AIS is a relatively new complex problem-solving approach when compared with other nature-inspired metaheuristics such as evolution strategies and genetic algorithms, its inherent characteristics of immune memory, vaccination, antibody self-recognition ability, and immunity diversity allow it to have a high level of flexibility and a good balance between global and local search (Bernardino and Barbosa, 2009a)

The process starts with an initial set of random solutions called antibodies. Each antibody in the pool represents the set of assigned PI-hubs and delivery times. The fitness of each solution is measured during each generation of an antibody, and solutions are selected for hypermutation, selection and cloning based on their fitness (computed using objective function values).

An antibody is represented as a bit string in which all the possible information regarding assigned PI-hubs and delivery times should be contained. The representation space χ is expressed as a Cartesian product of allele sets A^ξ . $\xi = L$ stands for forward flow and $\xi = E$ for reverse flow:

$$\chi = A^L \times A^E \quad (7.22)$$

The algorithm uses a 5-bit scheme $A^\xi = \{a_k^\xi a_j^\xi a_{t_1}^\xi a_i^\xi a_{t_2}^\xi; a_l^\xi \in \{0, 1\} / l = k, j, i, t_1, t_2\}$, so that for forward flow, each bit, if equal to 1, represents respectively a PI-hub k assigned to receive full PI-container from supplier j at time t_1 and to deliver to customer i in time t_2 . For reverse flow, each bit, if equal to 1, represents respectively a PI-hub k assigned to deliver empty PI-container to supplier j at time t_1 and receive empty container from customer i in time t_2 .

Later, each antibody is re-converted to binary variables $x_{jkt}^L, y_{ikt}^L, x_{jkt}^E$ and y_{ikt}^E to be able to solve the FMILP and then determine its related fitness F_χ computed with regard to the objective function OF_χ :

$$F_\chi = \frac{1}{OF_\chi} \quad (7.23)$$

Therefore, the probability, PS_χ , of selecting an antibody for cloning depends on its fitness. The probability PS_χ is defined as: $PS_\chi = \frac{F_\chi}{\sum_x F_\chi}$

As for exploration and exploitation processes, AIS uses Hyper Mutation (HM) and Receptor Editing (RE) mechanisms. Each operator conducts random perturbations on each gene to generate the next generation's population. The HM mechanism guarantees that the higher-fitness antibodies are hyper-mutated at a slower rate. The HM_χ the rate for an antibody χ is defined as $HM_\chi = e^{-\omega F_\chi}$ where ω is the decay control factor. Following hyper-mutation, a new population is generated. Antibodies are so reorganised based on the affinity evaluation.

Following the cloning and mutation operations, the RE mechanism is performed, which consists of removing a subset of the current antibodies (the worst $\phi\%$ percent of the population) and replacing them with newly generated antibodies (De Castro and Von Zuben, 1999). This mechanism allows exploring new search areas, which helps escape from local optima.

7.5.2 . AIS enhanced by Deep Q-learning

Although several studies have demonstrated the convergence of AISs to a global optimum, a Markov-chain analysis reveals poor convergence of the AIS algorithms (Bernardino and Barbosa, 2009a). Indeed, AIS converges slowly owing to the antibody's single-point random mutation, which means that a particular antibody chooses a gene bit and changes its value randomly to one of the other selectable values. Furthermore, because of other weak gene blocks, it cannot preserve any locally great gene blocks in some low-affinity antibodies. As a result, the speed of the search is slow.

In this paper, we use the same Deep Q-learning (DQ) technique thoroughly explained in Achamrah et al. (2021) to speed up the algorithm and enhance its performance. Indeed, metaheuristics (in our case, AIS) produce a large amount of data throughout their iterative search processes, which can be converted into explicit knowledge using Machine Learning models (Talbi, 2020). This information covers, for instance, the search's spaces, moves, recombination, local optima, and excellent and bad solutions. In this paper, DQ is used to help analyse data relative to movements and recombination that have been made to generate good solutions to the problem. The objective is to extract useful information from this data so that the AIS's search performance and speed may be improved.

The concept of Q-learning consists of placing the agent (i.e., antibody) in state-action pair sequences (i.e., mutation, -recombination), observing the rewards that ensue, and updating a table's predictions (called a Q-table) until the best action is accurately predicted. As a result, the "Q" stands for quality, which determines how effective a specific action is in earning a prospective reward. On the other hand, the process can be time-consuming since the amount of memory needed to save and update that table rises as the number of states increases, and the amount of time required to investigate each state to generate the requisite Q-table becomes thus impractical.

These Q-values are approximated using neural networks, which are referred to as DQ, to tackle this problem. Consequently, the state is given as an input, and the output is the Q-value of all potential actions. After the network has been trained, picking the best action requires comparing the potential rewards of each action and selecting the best one. Accordingly, DQ guides AIS in choosing the best actions to perform regarding the best moves for HM and RE.

To help speed up the AIS even further, all of the moves made so far are recorded. This "memory" is exploited so that the prior best actions are used every time the algorithm is applied to solve the model for a given instance, which happens to be similar to the ones already treated. The best moves are chosen based on how the new instance compares to the prior ones. The K-Nearest Neighbours technique is used to determine similarities (Achamrah et al., 2021).

Finally, a basic penalty method is used to respect the model's constraints in the AIS. Each antibody's feasibility is evaluated against the model's constraints during HM and RE procedures. If an infeasibility occurs, the value of the fitness functions of the corresponding antibody is penalised. Consequently, infeasible antibodies have a lower chance of making it into the following generation of antibodies.

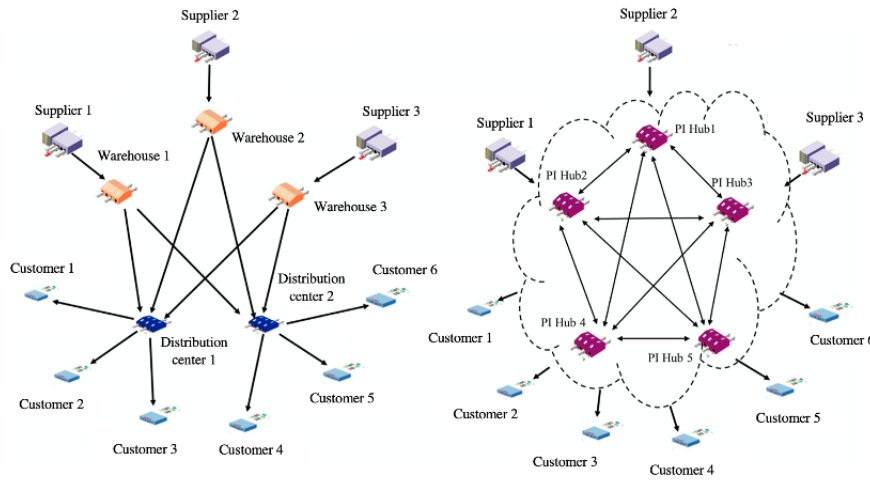


Figure 7.3: Current dominant distribution system in FMCG vs. PI for forward flows (source: [Yang et al. \(2017\)](#))

The algorithm stops when a time limit is reached, or no improvement in the quality of the solution is noted.

7.6 . Computational experiments

We compare the performance of our PI-supply chain to the classical supply chain described in Figure 7.3, which depicts forward flows in the distribution system in FMCG. The classical supply chain we address consists of suppliers from which products (manufactured by companies) are shipped using reusable transport items they own (e.g., pallets, boxes, .) to customer zones via the distribution centres to meet each customer's demand. In addition, neither direct shipment of products nor transshipment are allowed. Similarly, in the reverse flow, empty reusable items are collected from the customer zones and returned to their owner. All the production and maintenance processes (cleaning, recovering and disposal) are assumed to be performed at the suppliers regarding the type of returned items.

The performance ratio related to the total cost (TC) is computed as follows:

$$\mathcal{R} = \left(\frac{TC_{PI} - TC_{Classic}}{TC_{PI}} \right) 100 \quad (7.24)$$

The improvement in service levels (SL) is also computed as follows:

$$\mathcal{S} = \left(\frac{SL_{PI} - SL_{Classic}}{SL_{PI}} \right) 100 \quad (7.25)$$

7.6.1 . Experimental design

This section presents the experimental design for generating a set of instances since no benchmark instances were found in the PI literature with the same constraints and input/output data. As a result, we generate our data to be as realistic

Table 7.2: The model's input data

Parameter	Value
Number of customers	[5,200]
Number of PI-hubs	[5,20]
Number of suppliers	[5,40]
Number of PI-containers	[5,40]
Average demand	$U(20, 240)$
Standard deviation	$U(2, 144)$
Maximum inventory level of empty and loaded PI-containers at each customer i	$\nu_i \sum_{p,t} D_{pit} ; \nu_i = U(1, 3)$
Maximum inventory level of empty and loaded PI-containers at each supplier j	$\nu_j \sum_{p,i,t} D_{pit} ; \nu_j = U(1, 4)$
Maximum inventory level of empty and loaded PI-containers at each PI-hub k	$\nu_k \sum_{p,i,t} D_{pit} ; \nu_k = U(5, 10)$
Initial inventory level of a loaded PI-container p at customer i	$\nu_i D_{pi1} ; \nu_i = U(1, 2)$
Initial inventory level of an empty PI-container p at customer i	D_{pi1}
Initial inventory level of a loaded/empty PI-container p at supplier j	$\nu_j \sum_{p,i,t} D_{pit} ; \nu_j = U(5, 10)$
Initial inventory level of a loaded/empty PI-container p at PI-hub k	$\nu_k \sum_{p,i,t} D_{pit} ; \nu_k = U(5, 10)$
Inventory holding cost of loaded PI-container p per period	$U(0.02, 0.2)$
Inventory holding cost of empty PI-container p per period	$U(0.01, 0.1)$
production cost	$U(0.01, 0.04)$
New PI-container procurement cost	$U(10, 200)$
Lost sales cost	$100h_{pi}^L$
Time windows	$U(0, 5)$
Transportation cost from a node i to j with regard to the spacial coordinate X and Y generated in $U(0, 500)$. This concerns any level of the supply chain.	$\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} + 0.5$
Transshipment cost from PI-hub k to k'	$0.01(\sqrt{(X_k - X_{k'})^2 + (Y_k - Y_{k'})^2} + 0.5)$

as possible by following some of the standards for instances generated for the IRP and IRP for returnable transport items, namely demands, initial inventories and storage capacities (Lassinovskaia et al., 2017; Archetti et al., 2007, 2012; Coelho et al., 2014a). Our dataset, available upon request, is generated according to Table 7.2.

To evaluate the stability of the resolution approach, the tests are performed in 10 replications for each of the generated instances, and the average value of the objective function is presented. A time limit of 7200s is set for CPLEX to be able to validate the results obtained using the resolution approach. The analyses are also supported by one-tailed Wilcoxon signed-rank tests, which show that most observed differences are highly significant (p-value ≤ 0.001). Finally, all optimisation steps are carried out with a personal computer (MacBook Pro, macOS Big Sur, 3.3 GHz Quad-Core Intel Core i7 CPU with 8 GB of RAM) and with CPLEX 12.9, Python 3.7 and Pytorch.

7.6.2 . Parameters tuning of AIS and DQ

To enhance their performance, modern optimisation algorithms often require many parameters to be adjusted. The automatic algorithm configuration's immediate purpose is to discover the best optimiser parameter values automatically. Automatic algorithm setting can lead to new optimisation software design paradigms in the long run. The Irace package is a piece of software that automates a number of configuration operations (López-Ibáñez et al., 2016). Iterated racing processes, in particular, have been effectively employed to automatically configure a variety of state-of-the-art algorithms (López-Ibáñez et al., 2016). The iterated F-race algorithm and various adaptations and refinements are among the repeated racing

techniques implemented in Irace.

In this paper, a set of training instances representing the problem (80 instances with 5 to 50 customers, 5 to 10 suppliers and 5 to 10 PI-hubs) is used to choose the best algorithm configuration (see Table 7.3). The selected algorithm configuration can then be used to solve new instances of the same problem. Table 7.4 reports the parameters tuning for AIS and DQ.

Table 7.3: Tuning of AIS's parameters

Parameter	Range	Chosen values
Population size	[170,200]	200
Maximum number of iterations	[150,200]	200
HM probability	[0.31,0.34]	0.31
Selection probability	[0.47,0.54]	0.5
RE probability	[0.2,0.3]	0.2

Table 7.4: Tuning of DQ's parameters

Parameter	Value
Exploration rate	0.32
Discount rate	0.41
C	120

7.6.3 . Computational results

This section reports computational results on medium and large instances and sensitivity on unitcost.

7.6.3.1 . Results for medium and large instances

Table 7.5 report total costs computed for both the PI model (PI-M) and classical model (C-M) using CPLEX and our Algorithm. The set of instances under consideration consists of a number of customers N varying between 5 and 50; a number of suppliers S and PI-hubs K varying between 5 and 10 and a number of PI-containers P varying between 5 and 40. Table 7.5 also provides the CPU time in the second computed for PI-M and C-M. Table 7.6 provides for all instances under consideration the breakdown of costs for PI-M, namely: transportation (T), inventory at the customers (I-C), inventory at the suppliers (I-S), inventory at the PI-hubs (I-P), lost sales (LS), procurement (Pr), transshipment (Ts) and production (M). Table 7.6 also reports cost saving and service level improvement computed using CPLEX and our algorithm.

Table 7.5: Computational results on medium instances obtained using CPLEX and our algorithm

Instances					CPLEX				Algorithm				\mathcal{R}_{CPLEX} (%)	\mathcal{R}_{Algo} (%)	\mathcal{S}_{CPLEX} (%)	\mathcal{S}_{Algo} (%)
#	N	S	K	P	Total Cost		CPU (s)		Total Cost		CPU (s)					
					PI-M	C-M	PI-M	C-M	PI-M	C-M	PI-M	C-M				
1	5	5	5	5	8,794.0	15,444.1	242	210	9,401.2	16,358.8	1,039	385	43.06	42.53	30.65	48.81
2	5	5	5	10	18,473.2	32516.0	457	160	19,891.9	33,708.9	754	1,187	43.19	40.99	50.55	47.96
3	5	5	5	20	28,081.4	44,307.8	1,003	272	28,431.5	44,333.5	1,054	763	36.62	35.87	42.78	65.70
4	5	5	5	30	43,426.9	69,220.4	2,851	644	44,204.6	71,772.9	925	340	37.26	37.26	38.41	59.08
5	5	5	5	40	53,559.7	97,737.4	2,468	1,803	5,6945.5	104,911.8	1,001	521	45.20	45.72	60.12	48.47
6	5	5	10	5	12,829.6	18,986.8	2,013	1,498	14,059.5	19,448.4	1,147	463	32.43	27.71	64.04	55.08
7	5	5	10	10	16,416.4	26,155.5	2,488	2,720	16,628.0	23,480.5	975	856	37.24	29.18	41.06	26.10
8	5	5	10	20	32,512.3	57,259.1	2,914	2,053	33,562.8	58,139.2	723	677	43.22	42.27	37.88	55.51
9	5	5	10	30	44,121.6	81,345.9	2,552	2,657	46,243.7	84,089.1	852	654	45.76	45.01	41.39	29.46
10	5	5	10	40	48,981.0	88,185.0	2,972	2,743	52,951.1	94,819.5	730	311	44.46	44.16	29.08	31.92
11	5	10	10	5	11,515.1	21,075.8	1,202	2,337	11,788.4	21,700.9	806	864	45.36	45.68	26.16	46.63
12	5	10	10	10	16,803.7	28,774.8	3,137	1,193	17,529.4	29,133.9	523	656	41.60	39.83	44.09	47.04
13	5	10	10	20	35,071.6	68,427.3	3,643	7,097	37,677.1	68,814.2	458	707	48.75	45.25	40.99	41.13
14	5	10	10	30	51,159.8	73,729.4	3,077	3,077	52,223.9	73,883.8	394	700	30.61	29.32	43.15	55.60
15	5	10	10	40	69,059.1	121,055.5	4,385	4,807	71,731.7	122,833.3	1,079	961	42.95	41.60	57.58	46.20
16	10	10	10	5	16,850.3	31,762.5	3,034	2,669	17,638.8	33,360.6	932	708	46.95	47.13	31.58	43.44
17	10	10	10	10	25,439.2	38,256.2	2,883	7,178	26,750.3	36,326.6	832	1,186	33.50	26.36	62.88	40.59
18	10	10	10	20	41,450.1	59,971.3	6,833	6,571	43,839.4	59,774.1	823	1,192	30.88	26.66	37.16	54.65
19	10	10	10	30	64,592.8	88,010.2	1,508	4,617	67,924.9	96,397.0	520	398	26.61	29.54	53.53	50.94
20	10	10	10	40	82,709.0	148,545.9	4,332	6,436	86,553.8	150,465.9	369	1,029	44.32	42.48	44.77	64.28
21	20	10	10	5	40,022.6	70,378.6	2,387	2,911	41,027.8	76,191.1	1,155	886	43.13	46.15	24.94	26.59
22	20	10	10	10	54,618.6	85,311.7	5,678	2,862	55,682.8	88,531.8	672	1,149	35.98	37.10	51.27	42.88
23	20	10	10	20	85,995.3	130,593.8	6,064	6,773	88,207.2	134,041.5	1,166	791	34.15	34.19	54.70	54.14
24	20	10	10	30	122,095.2	198,578.3	1,507	6,364	124,500.9	214,544.8	775	1,171	38.52	41.97	65.37	57.95
25	20	10	10	40	166,125.3	274,701.7	5,030	4,261	167,262.4	220,674.3	580	375	39.53	24.20	38.71	40.06
26	30	10	10	5	74,311.9	109,580.7	3,882	7,043	75,628.4	110,201.9	420	1,179	32.19	31.37	64.46	29.45
27	30	10	10	10	112,564.5	206,602.5	4,320	5,494	117,416.2	211,236.5	914	681	45.52	44.41	59.72	66.15
28	30	10	10	20	398,623.3	767,668.9	7,200	7,200	183,338.0	267,244.5	663	669	48.07	31.40	40.52	41.57
29	30	10	10	30	380,238.8	584,237.2	7,200	7,200	271,971.8	404,941.5	1,038	455	34.92	32.84	42.80	60.90
30	30	10	10	40	840,315.9	1,209,253.2	7,200	7,200	355,656.7	612,341.3	400	1,121	30.51	41.92	50.56	55.56
31	40	10	10	5	334,506.8	699,251.3	7,200	7,200	161,498.2	222,234.5	821	1,166	52.16	27.33	39.10	46.10
32	40	10	10	10	235,585.1	383,937.2	7,200	7,200	247,969.2	370,952.0	655	384	38.64	33.15	66.31	52.58
33	40	10	10	20	946,823.0	1,943,387.2	7,200	7,200	397,985.0	612,674.0	1,187	1,199	51.28	35.04	43.05	36.89
34	40	10	10	30	2,373,723.7	3,196,527.2	7,200	7,200	595,835.5	960,457.1	769	783	25.74	37.96	44.23	61.61
35	40	10	10	40	1,867,767.6	4,339,149.1	7,200	7,200	814,733.0	1,116,062.0	718	919	56.96	27.00	57.93	51.61
36	50	10	10	5	849,863.2	1,343,747.1	7,200	7,200	351,513.9	667,145.3	1,034	1,178	36.75	47.31	33.48	52.62
37	50	10	10	10	1,118,547.8	1,494,275.0	7,200	7,200	523,683.8	740,729.7	856	502	25.14	29.30	24.54	31.14
38	50	10	10	20	2,241,369.7	3,658,178.7	7,200	7,200	863,703.1	1,607,835.2	404	1,006	38.73	46.28	27.23	29.93
39	50	10	10	30	3,445,654.3	4,546,979.9	7,200	7,200	1,288,021.0	2,269,171.0	710	1,071	24.22	43.24	44.31	31.82
40	50	10	10	40	3,621,136.9	5,964,566.2	7,200	7,200	1,750,128.7	3,072,910.9	833	1,086	39.29	43.05	26.11	30.57

From Tables 7.5, we can see that for all instances under consideration, and the proposed PI model helps reduce the total logistics cost compared to the classic inventory models with better service levels. The average total cost is reduced by 40%, and the average service level is increased by 44%. This improvement, as it can be noticed from Table 7.5, is only made possible thanks to the reduction of transportation, lost sales, inventory and procurement costs. Indeed, the classic inventory model results in significant inventory and transportation costs at all levels: suppliers, customers, warehouses and distribution centres. With the PI model, both forward and reverse flows are consolidated at the PI-hubs and thanks to the direct shipment transportation costs are reduced (as the truck fill rates are improved). Allowing transshipment between PI-hubs also reduces further inventory costs and better meets the customers' demand which systematically reduces lost sales. Finally, sharing empty PI-containers also reduces the cost of procurement of new containers.

Since CPLEX uses an exact method, Branch-and-Cut, it can fail to find exact solutions in a reasonable amount of time. Table 7.5 also reports the best results obtained by CPLEX and the proposed algorithm. From the table, we can see that the algorithm outperforms CPLEX significantly in solution quality and CPU times (given in seconds). The algorithm provides a better solution in all the test sets and can provide feasible solutions with an average gap of 4% for optimal solutions and can provide solutions from 28–62% better than those obtained by CPLEX for other feasible solutions, and this within only less than 16% the time spent by CPLEX.

To get more insight into the representiveness of the model and benefits of PI, extra experiments are conducted. We consider a set of customers N varying between 50 and 200, a number of suppliers varying between 20 and 40, a number of PI-hubs varying between 10 and 20 and finally, a number of PI-containers varying between 5 and 40. Table 7.7 summarizes obtained results and Table 7.8 reports the breakdown of costs.

Table 7.6: Breakdown of costs for medium size instances

Instances					T	I-C	I-S	I-P	LS	Pr	Ts	M
#	N	S	K	P								
1	5	5	5	5	4,172.4	976.9	1,720.0	117.8	932.5	299.0	50.9	1,131.7
2	5	5	5	10	5,777.9	3,040.7	7,784.8	361.9	805.9	329.0	190.6	1,601.2
3	5	5	5	20	6,026.0	4,929.8	11,856.3	901.7	1,150.6	88.0	111.3	3,368.0
4	5	5	5	30	4,167.7	8,522.6	23,935.6	1,364.2	968.6	246.0	240.5	4,759.4
5	5	5	5	40	4,120.6	13,223.5	29,547.6	1,938.9	779.5	317.0	330.1	6,688.3
6	5	5	10	5	3,453.3	2,081.9	6,826.1	239.0	197.3	251.0	251.0	760.0
7	5	5	10	10	3,605.4	2,260.2	7,346.6	373.1	1131.5	85.0	150.9	1,675.4
8	5	5	10	20	4,708.0	5,807.5	18,284.6	936.5	448.4	265.0	350.3	2,762.6
9	5	5	10	30	5,960.1	9,391.9	23,183.3	1,519.7	721.3	90.0	430.4	4,946.9
10	5	5	10	40	5,497.0	12,386.0	24,951.7	2,036.7	1,144.4	247.0	580.7	6,107.7
11	5	10	10	5	3,630.4	1,038.5	1,966.5	74.1	16.9	273.0	180.4	4,608.6
12	5	10	10	10	3,893.9	3,100.4	2,640.2	361.2	549.7	227.0	160.5	6,596.5
13	5	10	10	20	4,309.0	4,666.5	8,162.2	1,020.6	1,000.4	252.0	790.3	17,476.2
14	5	10	10	30	5,430.5	9,711.6	9,988.9	1,468.4	832.3	61.0	480.7	24,250.7
15	5	10	10	40	6,595.7	11,771.6	18,552.1	1,421.9	22.5	186.0	620.1	32,562.0
16	10	10	10	5	6,246.9	3,070.2	1,373.7	50.2	1748.4	260.0	500.8	4,388.7
17	10	10	10	10	6,975.7	5,253.5	3,483.6	432.7	2,064.9	32.0	728.0	7,779.9
18	10	10	10	20	7,160.3	9,741.5	6,300.9	1,137.1	3,014.4	308.0	932.0	15,245.3
19	10	10	10	30	8,641.6	15,723.6	17,360.0	1,478.4	2,945.7	271.0	1,049.2	20,455.6
20	10	10	10	40	8,680.4	19,543.7	17,892.2	2,252.9	3,735.8	84.0	1,450.9	32,913.8
21	20	10	10	5	13,365.1	6,859.8	3,112.0	110.1	6,954.5	55.0	1119.9	9,451.3
22	20	10	10	10	16,400.8	8,463.3	7,909.0	965.0	3,871.6	85.0	1,561.0	16,427.2
23	20	10	10	20	21,168.1	9,334.6	14,475.2	2,404.4	6,542.0	273.0	1,966.3	3,2043.7
24	20	10	10	30	24,394.5	10,286.8	37,032.9	3,192.1	2,017.6	452.0	2,211.8	44,913.2
25	20	10	10	40	28,769.9	12,990.5	38,282.6	4,970.6	3,911.8	36.0	3,063.3	75,237.7
26	30	10	10	5	28,200.5	14,617.7	6,591.0	231.3	2,771.0	424.0	2,443.9	20,349.1
27	30	10	10	10	33,988.6	18,911.3	17,856.2	2,185.4	5,223.2	177.0	3,530.0	35,544.5
28	30	10	10	20	41,770.7	21,843.7	31,690.0	5,130.0	6,798.9	79.0	4,330.3	71,695.5
29	30	10	10	30	47,040.5	24,407.9	82,504.0	7,322.2	7,168.3	519.0	4,680.6	98,329.4
30	30	10	10	40	57,369.6	28,499.4	83,497.8	10,621.8	1,134.8	199.0	6,503.0	167,831.3
31	40	10	10	5	62,198.1	31,387.8	14,074.4	516.4	3,119.5	628.0	5,215.4	44,358.6
32	40	10	10	10	78,272.0	36,528.8	37,802.0	4,773.2	6,348.0	363.0	8,098.9	75,783.4
33	40	10	10	20	95,420.6	45,117.4	72,308.6	11,703.8	127.1	596.0	9,757.7	162,953.8
34	40	10	10	30	121,582.1	55,683.9	182,472.5	16,682.2	1,604.8	4.0	10,389.0	207,417.1
35	40	10	10	40	136,596.3	68,832.5	189,473.3	22,841.8	2,763.9	142.0	14,586.5	379,496.7
36	50	10	10	5	134,060.6	66,415.2	30,776.8	1,127.0	6,411.3	302.0	11,793.0	100,627.9
37	50	10	10	10	161,096.7	83,769.5	80,067.7	10,788.4	4,721.4	74.0	17,299.8	165,866.3
38	50	10	10	20	198,039.3	98,506.3	158,937.9	26,899.8	340.9	474.0	22,402.5	358,102.4
39	50	10	10	30	221,544.6	123,583.0	419,044.4	35,797.9	355.0	327.0	23,860.4	463,508.7
40	50	10	10	40	246,920.3	149,586.1	412,940.0	50,136.3	5,105.3	550.0	31,633.9	853,256.7

Table 7.7: Computational results for large size instances

Instances #	Instances				Total cost		CPU (s)		\mathcal{R} (%)	\mathcal{S} (%)
	N	S	C	P	PI-M	C-M	PI-M	C-M		
1	50	20	10	5	671,367.52	1,691,846.16	2268	1118	60.32	54.52
2	50	20	10	10	1,039,769.34	3,177,535.10	1942	2048	67.28	62.59
3	50	20	10	20	1,792,640.23	7,439,456.95	1887	282	75.90	43.83
4	50	20	10	30	2,654,005.98	6,568,664.79	1536	1829	59.60	66.60
5	50	20	10	40	3,814,545.05	13,804,838.53	2322	2402	72.37	40.73
6	50	30	20	5	1,348,191.40	3,960,986.33	2143	1829	65.96	44.07
7	50	30	20	10	2,255,208.92	7,685,752.01	359	979	70.66	61.43
8	50	30	20	20	3,781,244.84	13,483,919.10	796	391	71.96	63.77
9	50	30	20	30	5,482,473.43	22,077,920.50	376	1936	75.17	57.13
10	50	30	20	40	8,301,365.96	19,400,292.25	1573	2301	57.21	51.27
11	50	40	20	5	2,923,717.12	8,379,373.27	1692	1032	65.11	60.81
12	50	40	20	10	4,953,843.82	10,962,856.37	1379	1457	54.81	42.47
13	50	40	20	20	8,255,497.02	24,295,927.73	1001	659	66.02	51.37
14	50	40	20	30	11,741,816.15	34,098,234.11	906	2462	65.56	53.68
15	50	40	20	40	18,081,740.82	47,048,689.61	2427	481	61.57	58.78
16	100	20	10	5	2,112,277.91	7,069,794.17	1982	1603	70.12	66.60
17	100	20	10	10	3,390,063.16	9,688,800.52	923	1624	65.01	44.37
18	100	20	10	20	5,538,778.94	12,268,395.35	1837	563	54.85	41.83
19	100	20	10	30	8,187,659.16	20,198,955.16	2496	2365	59.46	65.51
20	100	20	10	40	12,399,944.14	50,442,972.75	1628	2353	75.42	46.43
21	100	30	20	5	4,290,204.08	10,193,524.90	1903	2041	57.91	51.91
22	100	30	20	10	7,108,076.17	25,198,130.03	2342	1878	71.79	46.52
23	100	30	20	20	12,001,551.48	40,349,216.09	2389	631	70.26	47.83
24	100	30	20	30	17,588,891.03	59,696,696.17	1472	2208	70.54	63.82
25	100	30	20	40	27,100,967.78	63,931,182.98	1152	1149	57.61	58.71
26	100	40	20	5	9,135,031.33	34,183,287.23	889	750	73.28	63.37
27	100	40	20	10	14,472,340.30	35,326,982.68	1119	1605	59.03	56.43
28	100	40	20	20	26,341,895.76	59,216,581.67	819	1926	55.52	40.18
29	100	40	20	30	36,305,695.30	132,552,093.55	339	1811	72.61	36.95
30	100	40	20	40	55,951,896.60	134,564,311.33	1063	1945	58.42	64.78
31	150	20	10	5	6,574,526.47	18,093,096.83	1165	342	63.66	49.68
32	150	20	10	10	10,838,937.18	25,352,274.06	920	518	57.25	36.59
33	150	20	10	20	18,118,395.14	55,496,644.30	843	1048	67.35	65.79
34	150	20	10	30	25,918,218.40	84,908,083.47	2020	565	69.47	65.78
35	150	20	10	40	38,854,783.38	120,138,990.21	1809	2420	67.66	66.62
36	150	30	20	5	13,706,278.12	56,195,740.29	647	492	75.61	63.01
37	150	30	20	10	22,658,744.09	55,355,311.80	1807	678	59.07	52.80
38	150	30	20	20	40,250,927.86	152,873,024.03	1352	1849	73.67	58.59
39	150	30	20	30	56,319,419.69	180,447,420.67	1601	1442	68.79	38.11
40	150	30	20	40	84,401,006.65	261,136,714.57	673	325	67.68	64.37
41	150	40	20	5	29,168,808.92	64,579,742.94	2320	1083	54.83	64.52
42	150	40	20	10	48,742,090.03	170,451,088.82	1502	547	71.40	58.57
43	150	40	20	20	84,768,507.46	253,457,837.29	1827	1123	66.56	48.01
44	150	40	20	30	124,380,496.09	497,521,984.35	1925	2399	75.00	37.96
45	150	40	20	40	177,380,660.26	513,517,011.46	1832	719	65.46	52.15
46	200	20	10	5	21,401,105.86	64,631,339.71	1566	2204	66.89	60.89
47	200	20	10	10	34,378,834.61	113,793,942.55	1517	1593	69.79	56.31
48	200	20	10	20	57,739,495.77	152,547,747.82	1076	2277	62.15	45.95
49	200	20	10	30	82,079,254.80	318,221,270.85	2279	1020	74.21	66.92
50	200	20	10	40	120,615,566.37	452,067,142.75	1624	706	73.32	38.66
51	200	30	20	5	48,142,327.29	177,452,618.41	2303	727	72.87	52.53
52	200	30	20	10	73,276,442.80	198,945,542.21	691	1110	63.17	59.90
53	200	30	20	20	128,557,365.45	474,762,350.59	1147	1417	72.92	42.06
54	200	30	20	30	180,123,884.87	470,843,835.04	1052	1458	61.74	66.03
55	200	30	20	40	263,504,139.29	749,405,772.14	992	2339	64.84	42.91
56	200	40	20	5	106,838,732.82	409,619,701.63	410	962	73.92	42.68
57	200	40	20	10	160,136,038.60	457,508,662.27	689	1966	65.00	66.23
58	200	40	20	20	276,194,604.15	856,203,272.86	773	1267	67.74	38.31
59	200	40	20	30	388,525,032.88	1,568,864,082.77	1835	609	75.24	60.69
60	200	40	20	40	557,973,728.33	1,500,949,329.20	657	320	62.83	44.56

Table 7.8: Breakdown of costs for large size instances

Instances					T	I-C	I-S	I-P	LS	Pr	Ts	M
#	N	S	C	P								
1	50	20	10	5	286.87	69.91	65.02	2.41	134.96	6.08	12.82	228.12
2	50	20	10	10	358.82	95.26	177.12	23.02	106.80	6.67	19.66	359.12
3	50	20	10	20	433.23	103.51	360.14	58.52	7.38	8.04	24.01	805.18
4	50	20	10	30	487.72	136.41	894.73	75.74	7.63	9.72	26.86	1,022.82
5	50	20	10	40	560.54	157.40	1,032.00	107.44	111.00	8.20	35.11	1,913.75
6	50	30	20	5	606.36	95.37	142.35	3.05	141.11	7.79	27.51	465.62
7	50	30	20	10	831.41	121.70	424.71	29.10	132.45	8.44	42.87	796.84
8	50	30	20	20	920.99	121.33	733.79	65.31	9.38	10.62	54.58	1,874.62
9	50	30	20	30	1,026.92	179.32	1,899.38	82.33	8.74	11.18	58.92	2,224.42
10	50	30	20	40	1,295.05	164.10	2,199.45	121.18	140.76	9.96	83.92	4,427.56
11	50	40	20	5	1,326.16	103.85	303.43	3.30	174.43	10.81	64.79	1,111.19
12	50	40	20	10	1,947.59	153.19	954.01	35.59	158.42	8.99	102.12	1,752.18
13	50	40	20	20	2,035.88	138.63	1,669.19	77.34	11.15	10.93	128.70	4,194.81
14	50	40	20	30	2,226.41	229.16	4,232.33	98.69	8.98	12.09	140.69	4,802.43
15	50	40	20	40	3,039.93	214.70	4,980.94	127.47	186.98	13.24	182.70	9,522.58
16	100	20	10	5	870.87	217.50	199.06	7.41	430.38	19.38	39.16	758.46
17	100	20	10	10	1,200.18	302.29	555.74	73.56	357.03	21.35	63.75	1,172.83
18	100	20	10	20	1,348.66	317.69	1,128.78	197.43	22.66	25.67	78.25	2,442.27
19	100	20	10	30	1,517.16	447.23	2,698.17	228.19	23.17	32.82	87.19	3,176.87
20	100	20	10	40	1,696.87	494.41	3,451.94	342.73	366.15	26.88	115.47	6,271.27
21	100	30	20	5	1,782.44	237.68	420.86	9.26	521.60	23.72	93.92	1,721.81
22	100	30	20	10	2,563.76	341.26	1,300.87	100.37	377.52	23.45	150.13	2,627.86
23	100	30	20	20	3,180.62	430.86	2,365.02	275.29	30.66	34.95	176.95	5,537.83
24	100	30	20	30	3,497.50	556.63	5,564.86	285.62	28.62	43.21	182.95	7,458.09
25	100	30	20	40	3,840.65	606.76	7,930.18	439.47	417.62	34.67	238.75	14,010.08
26	100	40	20	5	3,861.50	260.12	948.33	12.85	574.66	25.76	224.30	3,801.59
27	100	40	20	10	5,482.94	351.49	2,630.08	138.59	415.94	25.44	347.42	5,495.97
28	100	40	20	20	7,004.64	535.39	5,286.76	375.84	34.35	35.18	369.55	12,734.51
29	100	40	20	30	7,386.26	666.76	11,671.91	333.43	35.13	56.49	416.88	15,773.94
30	100	40	20	40	8,045.96	662.22	17,148.70	489.06	419.50	43.93	539.45	29,022.16
31	150	20	10	5	2,677.17	712.83	654.75	24.84	1,459.70	61.83	123.21	2,318.44
32	150	20	10	10	3,831.39	979.58	1,791.55	236.75	1,087.78	64.24	202.24	3,732.11
33	150	20	10	20	4,418.19	1,016.89	3,817.43	651.54	69.07	83.87	242.36	7,888.05
34	150	20	10	30	4,605.34	1,464.43	8,757.82	731.75	77.72	103.90	279.64	9,975.25
35	150	20	10	40	5,333.15	1,520.56	11,019.46	1,111.28	1,205.97	82.19	369.09	19,417.86
36	150	30	20	5	5,752.72	776.14	1,318.47	32.67	1,730.82	76.06	246.56	5,501.92
37	150	30	20	10	9,046.71	1,042.89	3,793.83	307.84	1,403.12	64.74	419.84	7,981.49
38	150	30	20	20	10,389.94	1,281.98	8,279.24	797.65	81.32	99.60	506.62	18,895.82
39	150	30	20	30	11,040.90	1,483.07	19,128.84	994.66	92.32	115.20	636.91	22,919.74
40	150	30	20	40	11,049.74	2,054.72	22,982.19	1,384.93	1,270.55	99.26	798.09	46,030.82
41	150	40	20	5	12,775.65	780.21	2,672.78	35.31	1,910.33	98.14	524.63	12,280.19
42	150	40	20	10	19,716.59	1,229.92	9,081.03	379.88	1,572.66	65.34	886.15	17,381.61
43	150	40	20	20	23,387.66	1,300.07	17,520.77	1,032.19	93.73	114.05	1,120.41	40,293.26
44	150	40	20	30	25,277.03	1,673.22	44,871.85	1,104.40	125.19	122.03	1,528.38	49,803.46
45	150	40	20	40	23,572.75	2,284.11	47,275.96	1,598.34	1,623.44	114.49	1,721.29	100,812.09
46	200	20	10	5	8,821.35	2,359.63	2,071.37	81.89	4,643.80	199.60	413.12	7,449.50
47	200	20	10	10	11,953.93	2,999.22	5,970.74	733.68	3,438.93	208.66	652.17	11,856.99
48	200	20	10	20	14,506.50	3,317.06	11,808.68	2,180.60	233.74	264.19	779.15	24,883.09
49	200	20	10	30	14,553.35	4,770.53	28,693.34	2,264.48	255.25	333.56	871.62	30,592.11
50	200	20	10	40	17,442.91	5,116.87	34,442.97	3,719.67	3,686.65	264.68	1,245.75	58,379.02
51	200	30	20	5	20,243.31	4,997.80	4,714.01	171.04	10,641.13	420.24	847.52	16,737.77
52	200	30	20	10	24,056.10	6,113.61	12,966.36	1,556.28	7,080.28	490.03	1,327.89	26,759.08
53	200	30	20	20	30,461.18	7,233.01	26,348.71	4,407.91	536.26	631.98	1,706.70	57,767.33
54	200	30	20	30	31,044.03	10,769.53	64,193.32	5,344.52	562.20	784.66	2,090.02	65,897.24
55	200	30	20	40	36,839.59	12,086.04	81,845.46	7,567.19	8,608.44	559.82	2,665.83	121,931.60
56	200	40	20	5	44,276.17	11,256.14	11,173.81	386.62	24,415.75	938.76	1,915.88	36,866.94
57	200	40	20	10	48,206.50	14,273.33	28,625.05	3,253.81	14,500.20	1,012.80	2,742.56	62,007.48
58	200	40	20	20	70,531.04	14,669.49	57,247.32	9,757.52	1,260.86	1,323.55	3,708.78	118,955.64
59	200	40	20	30	62,960.09	21,731.72	142,269.10	12,715.74	1,177.23	1,628.17	4,294.50	142,924.53
60	200	40	20	40	84,184.73	24,371.25	166,886.99	16,302.91	20,116.20	1,129.61	5,521.12	259,557.01

* All costs except LS are given on a x1000 scale

From Tables 7.7 and 7.8, we can observe the positive impact of the PI model on the logistics efficiency compared to the classical model. This implies that exploiting the high level of integration, flexibility and openness of PI can help firms achieve significant cost savings. That is, along with the increased flexibility of the supply chain network, cost benefits follow. Indeed, in the traditional model, frequent transportation is necessary to meet as much as possible demands and send empty containers back to the suppliers, increasing transportation costs. For holding, lost sales and procurement costs, the traditional supply chain network is a multi-echelon hierarchical structure, independent and fragmented, and thus,

the transportation is inefficient. That is, it is impossible to rapidly and efficiently satisfy demand. Therefore, the holding and lost sale costs are relatively high compared to the PI model. This also implies that each supplier needs to buy more containers to guarantee the continuity of production and supplies at its level. Referring to the parameters α and β , we can also notice that under a significant network scale, PI can more adequately exploit the inter-connection and synergy among the entities in the supply chain network to consolidate the fragmented and overlapping transportation flow and thus reduce its relative cost. That is, a larger network scale better highlights the advantages of distribution flexibility. Hence, the contributions of transportation cost savings to the total cost savings increase with the network scale. We can also observe that transshipment between PI-hubs increases with the number of suppliers and customers since such policy is used as a sourcing means, reducing transportation, lost sales and inventory costs.

7.6.3.2 . Sensitivity analysis on unit costs

Sensitivity analysis is conducted on the unit cost of inventory at the suppliers, transportation and production due to their significant contributions to the total costs. Table 7.9 reports the two scenarios under study. Figure 7.4 depicts the variation of the total cost of PI and classical model for various unit cost of transportation, Figure 7.5 for the unit cost of production and Figure 7.6 depicts their variation for various unit cost of inventory holding.

Table 7.9: Unit costs value

Scenario	Unit cost under study	Value
Scenario 1	Unit cost of transportation	$U(0.5, \alpha) ; \alpha = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$
Scenario 1	Unit cost of inventory	$U(0, 1, 1)$
Scenario 2	Unit cost of production	$U(0.01, 0.2)$

From Figure 7.4, we can notice that when the transportation cost is smaller, a lower cost is incurred as each level can quickly achieve certain flexibility in its operations by increasing the number of delivery and pickup, even in a classical supply chain network. Thus, the advantages of PI weaken. However, as the unit cost of transportation increases, the advantage of flexibility is gradually highlighted. Therefore, PI becomes more preferable with a higher unit cost of transportation. Moreover, from Figures 7.5 and 7.6, we can notice that as the units cost of production and inventory increase, the predominance of PI may weaken (as the difference between total costs is not notable, compared to the other settings). The reason may be that the advantages of PI mainly depend on its flexibility compared to classical configurations. In addition, this flexibility mainly includes flexible production, inventory, and distribution in practice. Thus, the advantage of the production flexibility of PI weakens with a higher inventory and production cost. Allowing transshipment between suppliers may qualify, as the literature has proved to reduce inventory costs at their level. Indeed, when transshipment is permitted, it offers several advantages: suppliers receiving the quantity latterly transshipped can satisfy even more demand and consequently reduce lost sales. Customers from which

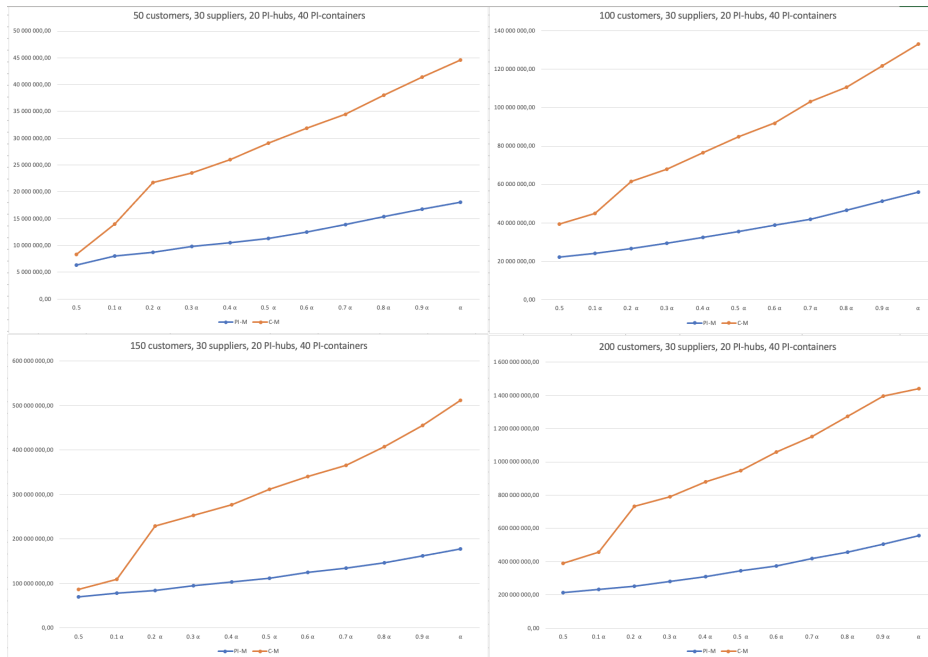


Figure 7.4: Variation of total cost for PI-M and C-M for various unit cost of transportation

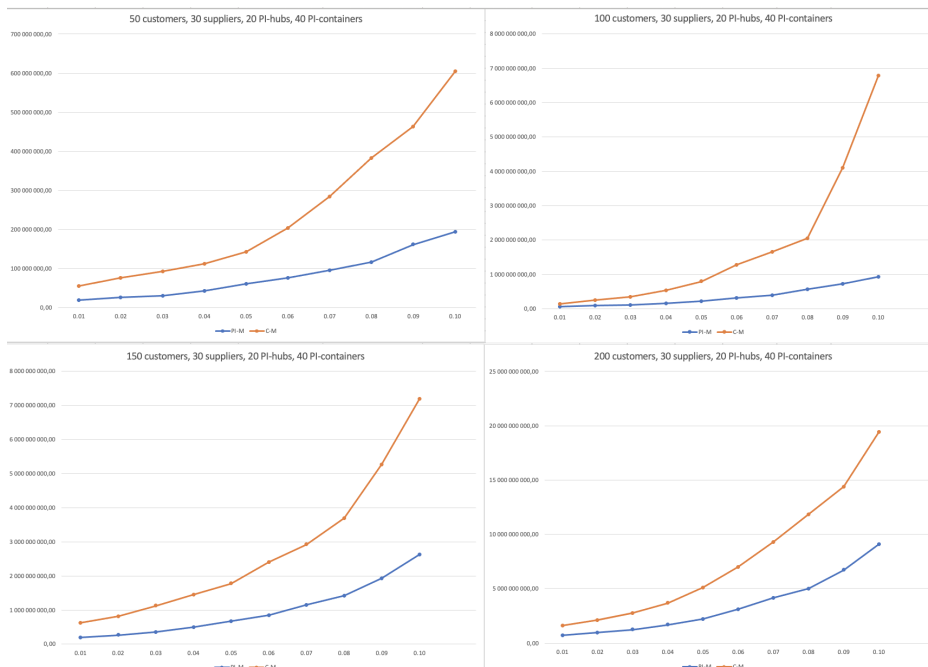


Figure 7.5: Variation of total cost for PI-M and C-M for various unit cost of production

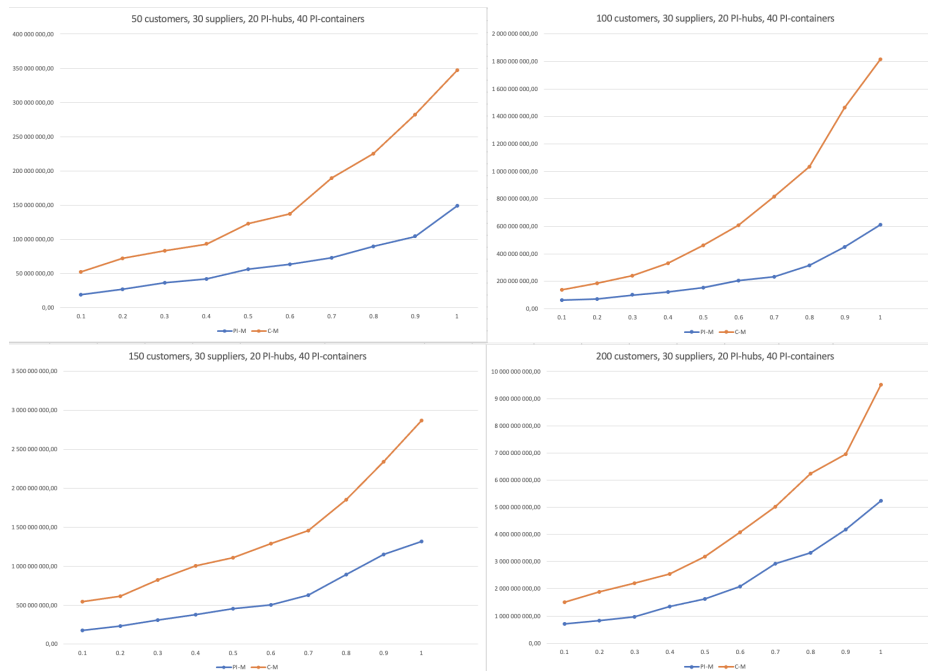


Figure 7.6: Variation of total cost for PI-M and C-M for various unit cost of inventory

the transshipment is carried out can lower their inventory holding costs (preventing thus idle stock).

7.7 . Conclusions & perspectives

This paper investigates the integration of forward and reverse networks in the interconnected logistics services in PI. A three-level PI-supply chain is examined in which a set of suppliers deliver their products using a set of PI-containers to customers. Direct and reverse flows of the PI-containers are consolidated at the level of a set of PI-hubs. This paper focuses more on the inventory routing model of reusable PI-containers exploiting the PI concept, which is new to the literature. Secondly, the paper considers inventory sharing between the PI-hubs, including empty and loaded PI-containers. Thirdly, deliveries from suppliers to the PI-hubs and from PI-hubs to the customers are considered to be performed within time windows. And finally, a resolution approach based on the hybridisation of mathematical modelling, artificial immune system algorithm and deep Reinforcement Learning is proposed to handle the combinatorial complexity of the problem at hand.

Computational experiments highlight that exploiting the high level of integration, flexibility and openness of PI can help firms achieve significant cost savings. That is, along with the increased flexibility of the supply chain network, cost benefits follow. They stress that PI can be more efficient under a larger network scale

and better highlights the advantages of distribution flexibility. Results also show the benefit of promoting transshipment in the overall supply chain. On the other hand, sensitivity analysis of unit cost highlights the limit of the PI model depending on the unit cost of transportation, inventory and production cost. As for the resolution approach, results show that it significantly outperforms CPLEX in solution quality and CPU times.

Further extensions of the present work may include investigation of new approaches to manage disruptions such as truck delays, customers changing or cancelling orders at the last minute, routing in PI-hubs, PI-hubs suddenly becoming unserviceable, etc. Also, the decisions related to fleet composition and fuel consumption alongside filling rate are to be considered in future work. Furthermore, it could be interesting to investigate multi-modal freight for delivery and pickup in the PI context. Finally, using the proposed resolution approach to solve other variants of the problem strengthens the present analysis and generalises findings that can be applied to more real and complex supply chains.

8 - General conclusions

The research topic of this thesis was motivated by real-world applications and identified gaps in the literature. Through several scientific articles, we tackled the critical issue of inventory sharing in distribution networks to assure product availability, timeliness, and consistency of delivery.

When dealing with distribution networks with different customers supplied from one or multiple warehouses, managers are asked to secure supplies and sidestep shortages as much as possible. The analysis of real-life applications indicates a special requirement for suppliers and customers regarding inventory management, material transportation, and product delivery. Managers must continuously perform a tricky balancing act, swinging between the urge to lower inventory levels to reduce costs and the requirement to avoid shortages to maintain high service levels.

The scope of our research is yet defined around partnerships, cooperation, and collaboration. We mainly focused on the inventory routing problem since inventory and transportation are the most critical issues of the logistics system and are the two main drivers that provide value to customers.

In all the presented papers, we proposed a new approach for supply chain players ready to collaborate to lower costs while maintaining a high level of customer service and avoiding stock-outs as much as possible. We designed a new inventory and routing management policy promoting multi-sourcing options to mitigate shortages. This policy combines regular shipment from manufacturer to customers and inventory sharing through LT among customers. The use of product substitutions is also investigated.

Each of the five articles comprehensively summarises the findings and limitations. Other broad concerns, as well as some of the research limitations and future study ideas, are discussed here.

From a mathematical perspective, we built our mathematical formulations on classical IRP models, which we adjusted and expanded to account for new limitations. The peculiarities of lateral transshipment, pickups and deliveries operations coupled with routing optimisation, as well as product substitutions, were brought to the fore by these new formulations. The formulations also capture different customer demand and supply chain settings, namely centralised and decentralised decision-making and reverse logistics.

To assess the benefits of promoting inventory sharing and products substitution within constrained and complicated configurations of real-world supply chains, the results obtained in the first and second articles (Chapters 2 and 3) were meant to serve as a basis for the elaboration of the rest of articles. Also, suitable resolution methods are designed using, as much as possible, the knowledge built upon mathematical formulation and hybridisation of sophisticated metaheuristics.

The results have shown that inventory sharing and substitutions may present a vital input for an effective performance appraisal within ever-volatile, complex,

uncertain and ambiguous environments. We examined the distribution of fashion items, perishable products, and spare parts for finished goods under centralised and decentralised settings and the management of returnable transport items in the automotive and food industries in closed-loop supply chains.

For each configuration, we quantified the impact of promoting this multi-sourcing on the overall performance of the supply chain. We demonstrated that these two options could help reduce costs and increase service levels through extensive experiments using either a literature benchmark or a randomly generated dataset. Also, we highlighted that the recourse to them is not only to mitigate shortages for finished products and reduce procurement costs for RTIs, but it might be a good practice to reduce inventory and transportation costs in deterministic or stochastic contexts. Regarding the combinatorial complexity, it is undeniable that capturing critical elements of supply chains increases the complexity of the developed models.

Several resolution approaches were used, evaluated, and enhanced throughout the process. To capture subtly and holistically all decisions related to the multi-sourcing, inventory management and vehicle routing, the first step was the hybridisation of the mathematical model, strengthened by a set of well-known valid inequalities and two sophisticated metaheuristics namely GA and SA. The proposed metaheuristic was tested on a set of benchmark instances and proved to be very competitive compared to well-known and exact algorithms in terms of run time. This also allows us to assess the limitation of GA regarding its scalability. Under stochastic context, we used sample average approximation approach. We also highlighted its advantages and limitations regarding the generation of low-quality solutions with a high computational burden. This initiated us to brainstorm solutions to how we could simultaneously build on the developments already made and bypass the so-called limitations. From this stems the idea of using RL technique. Indeed, since random searching leads to slow evolution and weak convergence efficiency, environmental feedback signals and the updated action policy of the deep Q-learning method are used to construct an algorithm with strong self-adaptability and goal-driven performance.

The promising results from stochastic and dynamic IRP encourage us to use this approach to solve the models we developed for more complex settings. We also compared it with pure metaheuristics and enhanced AIS. This comparison allowed us to assess the limits of our approach. Results showed that AIS improved with RL performs GA owing to its mechanisms, the speed and the accuracy of its immune response. Again, enhanced AIS is retained and used to solve a model developed for the PI setting.

Developing an inventory sharing modelling framework endowed with solving capabilities allowed us to capture different sharing strategies and investigate their potential benefits for the players. On the other hand, the more effectively we try to model real supply chain configurations, the more complex the models become. Results also highlight that no standard approach is 100% reliable and efficient for any problem and distribution network. Furthermore, ensuring better performance requires continuous improvement and upgrading of the developed algorithms.

Corroborating with this, our results indicated that, despite the attractiveness of inventory sharing and substitution, trade-offs on unit costs are significant to safeguard cost reduction. Hence, other multi-sourcing options can be considered, such as 3D printing. It would also be interesting to conducting more in-depth studies of organisational conditions and collaboration mechanisms in decentralized supply chains under which sharing would be beneficial to all involved players.

In addition, apart from detailed limitations displayed within each of the articles, all the developments (models and resolution approaches) could be of great insight regarding the relevance and ease of implementation if tested on a real dataset. Moreover, considering inventory back-ordering instead of lost sales would be interesting given its importance in some supply chain contexts (e.g. spare parts, RTIs, etc.). Furthermore, it would be interesting to conduct extra experiments to support the findings, such as sensitivity analysis on some strategic costs and analysing of the impact of demand uncertainty/

In this line of thinking, it would also be convenient to explicitly adjust the models designed and developed in line with the expectations of sustainable development goals of the United Nations, particularly the goal of 9 that aims to build resilient, inclusive, and sustainable industrialisation; and the goal 12 that focuses on fostering sustainable consumption and production patterns. Integrating some environmental indicators such as CO₂ emissions and energy consumption would also be interesting.

The use of hybrid modes for shared transportation could be an interesting way to extend the present work to highlight the use of zero-emission vehicles for last-mile deliveries in Physical Internet settings, for instance. Moreover, sustainability should be considered an intrinsic feature in the design of future models.

An interesting perspective for future work would be to investigate sharing other resources such as vehicles, machines and production or warehousing facilities and examine various collaboration mechanisms and negotiation protocols.

Finally, RL has been widely applied to various optimisation problems in many domains. However, it may be interesting to investigate the deep reinforcement learning limitation regarding its inability to question or rationalise the information provided.

Bibliography

- Abdelmaguid, T. F. (2004). *Heuristic approaches for the integrated inventory distribution problem*. PhD thesis, University of Southern California.
- Abdelmaguid, T. F., Dessouky, M. M., and Ordóñez, F. (2009). Heuristic approaches for the inventory-routing problem with backlogging. *Computers & Industrial Engineering*, 56(4):1519–1534.
- Abouee-Mehrzi, H., Berman, O., and Sharma, S. (2015). Optimal Joint Replenishment and Transshipment Policies in a Multi-Period Inventory System with Lost Sales. *Operations Research*, 63(2):342–350.
- Absi, N., Dautère-Pérès, S., Kedad-Sidhoum, S., Penz, B., and Rapine, C. (2013). Lot sizing with carbon emission constraints. *European Journal of Operational Research*, 227(1):55–61.
- Accorsi, R., Baruffaldi, G., Manzini, R., and Pini, C. (2019). Environmental Impacts of Reusable Transport Items: A Case Study of Pallet Pooling in a Retailer Supply Chain.
- Achamrah, F. E., Bouras, A., Riane, F., and Darmoul, S. (2019). Returnable Transport Items Management: A New Approach to Sidestep Shortage. In *7th International Conference IEEE on Advanced Logistics and Transport*.
- Achamrah, F. E., Riane, F., and Aghezzaf, E.-H. (2022a). Bi-level programming for modelling inventory sharing in decentralized supply chains. *Transportation Research Procedia*, 62:517–524. 24th Euro Working Group on Transportation Meeting.
- Achamrah, F. E., Riane, F., Bouras, A., and Sahin, E. (2020). Collaboration Mechanism for Shared Returnable Transport Items in Closed Loop Supply Chains. 9th International Conference on Operations Research and Enterprise Systems.
- Achamrah, F. E., Riane, F., Di Martinelly, C., and Limbourg, S. (2022b). A matheuristic for solving inventory sharing problems. *Computers & Operations Research*, 138:105605.
- Achamrah, F. E., Riane, F., and Limbourg, S. (2021). Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning. *International Journal of Production Research*, 0(0):1–18.
- Achamrah, F. E., Riane, F., and Limbourg, S. (2022c). Spare parts inventory routing problem with transshipment and substitutions under stochastic demands. *Applied Mathematical Modelling*, 101:309–331.

- Achamrah, F. E., Riane, F., Sahin, E., and Limbourg, S. (2022d). An artificial-immune-system-based algorithm enhanced with deep reinforcement learning for solving returnable transport item problems. *Sustainability*, 14(10).
- Achetoui, Z., Mabrouki, C., and Mousrij, A. (2019a). A review of spare parts supply chain management. *Jurnal Sistem dan Manajemen Industri*, 3:67.
- Achetoui, Z., Mabrouki, C., and Mousrij, A. (2019b). Performance measurement system for automotive spare parts supply chain: A categorization. *Journal of Transportation and Logistics*, pages 31–50.
- Adams, D., Oh, D.-H., Kim, D.-W., Lee, C.-H., and Oh, M. (2021). Deep reinforcement learning optimization framework for a power generation plant considering performance and environmental issues. *Journal of Cleaner Production*, 291:125915.
- Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014). Formulations and Branch-and-Cut Algorithms for Multivehicle Production and Inventory Routing Problems. *INFORMS Journal on Computing*, 26(1):103–120.
- Agostino, D., Arnaboldi, M., and Lema, M. D. (2021). New development: COVID-19 as an accelerator of digital transformation in public service delivery. *Public Money & Management*, 41(1):69–72.
- Ahmadian, S. (2017). Approximation Algorithms for Clustering and Facility Location Problems.
- Alegre, J., Laguna, M., and Pacheco, J. (2007). Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. *European Journal of Operational Research*, 179(3):736–746.
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., and Asari, V. K. (2019). A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, 8(3).
- Amirtaheri, O., Zandieh, M., and Dorri, B. (2017). A bi-level programming model for decentralized manufacturer-distributor supply chain considering cooperative advertising. *Scientia Iranica*, 25.
- Andersson, H., Christiansen, M., and Fagerholt, K. (2011). The Maritime Pickup and Delivery Problem with Time Windows and Split Loads. *INFOR: Information Systems and Operational Research*, 49:79–91.
- Anupindi, R., Bassok, Y., and Zemel, E. (2001). A general framework for the study of decentralized distribution systems. *Manufacturing & Service Operations Management*, 3(4):349–368.

- Archetti, C., Bertazzi, L., Hertz, A., and Speranza, M. G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1):101–116.
- Archetti, C., Bertazzi, L., Laporte, G., and Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391.
- Archetti, C., Christiansen, M., and Grazia Speranza, M. (2018). Inventory routing with pickups and deliveries. *European Journal of Operational Research*, 268(1):314–324.
- Archetti, C. and Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246.
- Archetti, C., Speranza, M. G., Boccia, M., Sforza, A., and Sterle, C. (2020). A branch-and-cut algorithm for the inventory routing problem with pickups and deliveries. *European Journal of Operational Research*, 282(3):886–895.
- Arikan, E. and Silbermayr, L. (2018). Risk pooling via unidirectional inventory transshipments in a decentralized supply chain. *International Journal of Production Research*, 56(17):5593–5610.
- Atan, Z., Snyder, L. V., and Wilson, G. R. (2018). Transshipment policies for systems with multiple retailers and two demand classes. *OR Spectrum*, 40(1):159–186.
- Avci, M. and Topaloglu, S. (2015). An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries. *Computers & Industrial Engineering*, 83:15–29.
- Avella, P., Boccia, M., and Wolsey, L. A. (2018). Single-period cutting planes for inventory routing problems. *Transportation Science*, 52(3):497–508.
- Axsäter, S. (2003). A new decision rule for lateral transshipments in inventory systems. *Management Science*, 49(9):1168–1179.
- Azadeh, A., Elahi, S., Farahani, M. H., and Nasirian, B. (2017). A genetic algorithm-taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers & Industrial Engineering*, 104:124 – 133.
- Babae Tirkolaee Erfan Goli, A. F. A. S. M. W. G.-W. (2020). Multi-Objective Optimization for the Reliable Pollution-Routing Problem with Cross-Dock Selection using Pareto-based Algorithms. *Journal of Cleaner Production*, page 122927.
- Bai, C., Quayson, M., and Sarkis, J. (2021). COVID-19 Pandemic Digitization Lessons for Sustainable Development of Micro-and Small-Enterprises. *Sustainable Production and Consumption*.

- Baker, B. M. and Ayechev, M. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5):787 – 800.
- Ballot, E. and Fontane, F. (2010). Reducing transportation co2 emissions through pooling of supply networks: perspectives from a case study in french retail chains. *Production Planning & Control*, 21(6):640–650.
- Ballot, E., Montreuil, B., and Meller, R. D. (2014). The physical internet: The network of logistics networks. null.
- Banerjee, A., Burton, J., and Banerjee, S. (2003). A simulation study of lateral shipments in single supplier, multiple buyers supply chain networks. *International Journal of Production Economics*, 81:103–114.
- Banholzer, D., Fliege, J., and Werner, R. (2019). On rates of convergence for sample average approximations in the almost sure sense and in mean. *Mathematical Programming*, pages 1–39.
- Bard, J. F. and Falk, J. E. (1982). An explicit solution to the multi-level programming problem. *Computers & Operations Research*, 9(1):77–100.
- Battarra, M., Cordeau, J.-F., and Iori, M. (2014). Chapter 6: pickup-and-delivery problems for goods transportation. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 161–191. SIAM.
- Batth, V. (2021). Toyota motor corporation: Just in time (jit) management strategy or beyond? *Journal of Case Research*, 12(1).
- Beiderbeck, D., Deradjat, D., and Minshall, T. (2018). The Impact of Additive Manufacturing Technologies on Industrial Spare Parts Strategies.
- Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., Mack, R. G., and Prutzman, P. J. (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23.
- Bello, I., Pham, H., Le, Q., Norouzi, M., and Bengio, S. (2016). Neural Combinatorial Optimization with Reinforcement Learning.
- Ben-Ayed, O. (1988). Bilevel linear programming : analysis and application to the network design problem.
- Benjamin, G. (2020). *Vehicle Routing Problem for Multi-Product Cross-Docking*.
- Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 15:1–31.

- Bernardino, H. S. and Barbosa, H. J. (2009a). Artificial immune systems for optimization. In *Nature-Inspired Algorithms for Optimisation*, pages 389–411. Springer.
- Bernardino, H. S. and Barbosa, H. J. C. (2009b). Grammar-Based Immune Programming for Symbolic Regression BT - Artificial Immune Systems. pages 274–287, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Bertazzi, L., Bosco, A., Guerriero, F., and Laganà, D. (2013). A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies*, pages –.
- Bertazzi, L. and Speranza, M. G. (2002). Continuous and discrete shipping strategies for the single link problem. *Transportation Science*, 36(3):314–325.
- Bortolini, M., Galizia, F. G., Mora, C., Botti, L., and Rosano, M. (2018). Bi-objective design of fresh food supply chain networks with reusable and disposable packaging containers. *Journal of Cleaner Production*, 184:375–388.
- Bouza Allende, G. and Still, G. (2012). Solving bilevel programs with the kkt-approach. *Mathematical Programming*, 138.
- Boylan, J. E., Syntetos, A. A., and Karakostas, G. C. (2008). Classification for forecasting and stock control: a case study. *Journal of the operational research society*, 59(4):473–481.
- Burns, L. D., Hall, R. W., Blumenfeld, D. E., and Daganzo, C. F. (1985). Distribution strategies that minimize transportation and inventory costs. *Operations research*, 33(3):469–490.
- Burton, J. and Banerjee, A. (2005). Cost-parametric analysis of lateral transshipment policies in two-echelon supply chains. *International Journal of Production Economics*, 93:169–178.
- Butt, A. S. (2021). Strategies to mitigate the impact of COVID-19 on supply chain disruptions: a multiple case analysis of buyers and distributors. *The International Journal of Logistics Management*.
- Campbell, A., Clarke, L., Kleywegt, A., and Savelsbergh, M. (1998). *The Inventory Routing Problem*, pages 95–113. Springer US, Boston, MA.
- Carrasqueira, P., Alves, M. J., and Antunes, C. H. (2017). Bi-level particle swarm optimization and evolutionary algorithm approaches for residential demand response with different user profiles. *Information Sciences*, 418-419:405–420.
- Cavagnini, R., Bertazzi, L., and Maggioni, F. (2018). *A Two-Stage Stochastic Model for Distribution Logistics with Transshipment and Backordering: Stochastic Versus Deterministic Solutions: ODS, Taormina, Italy, September 10–13, 2018*, pages 131–140.

- Chen, H.-K., Chou, H.-W., Hsueh, C.-F., and Yu, Y.-J. (2014). The paired many-to-many pickup and delivery problem: an application. *TOP*, 23.
- Chen, P., Guo, Y., Lim, A., and Rodrigues, B. (2006). Multiple crossdocks with inventory and time windows. *Computers & Operations Research*, 33:43–63.
- Chiang, C.-W., Lee, W.-P., and Heh, J.-S. (2010). A 2-Opt based differential evolution for global optimization. *Applied Soft Computing*, 10(4):1200–1207.
- Chiou, C.-C. (2008). *Transshipment Problems in Supply Chain Systems: Review and Extensions*.
- Chitsaz, M., Cordeau, J.-F., and Jans, R. (2019). A unified decomposition matheuristic for assembly, production, and inventory routing. *INFORMS Journal on Computing*, 31(1):134–152.
- Christiansen, M. (1999). Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science*, 33(1):3–16.
- Christiansen, M. and Fagerholt, K. (2014). *Chapter 13: Ship Routing and Scheduling in Industrial and Tramp Shipping*, pages 381–408.
- Christiansen, M., Fagerholt, K., Flatberg, T., Haugen, Ø., Kloster, O., and Lund, E. H. (2011). Maritime inventory routing with multiple products: A case study from the cement industry. *European Journal of Operational Research*, 208(1):86–94.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483.
- Chrysochoou, E., Ziliaskopoulos, A., and Lois, A. (2015). An exact algorithm for the stochastic inventory routing problem with transshipment.
- Cobb, B. (2016). Inventory control for returnable transport items in a closed-loop supply chain. *Transportation Research Part E: Logistics and Transportation Review*, 86:53–68.
- Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2012a). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, 24:270–287.
- Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2012b). The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537–2548.
- Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2014a). Heuristics for dynamic and stochastic inventory-routing. *Computers & Operations Research*, 52:55–67.

- Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2014b). Thirty years of inventory routing. *Transportation Science*, 48(1):1–19.
- Coelho, L. C. and Laporte, G. (2013a). A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research*, 51(23-24):7156–7169.
- Coelho, L. C. and Laporte, G. (2013b). The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, 40(2):558–565.
- Coelho, L. C. and Laporte, G. (2014). Improved solutions for inventory-routing problems through valid inequalities and input ordering. *International Journal of Production Economics*, 155:391–397.
- Colson, B., Marcotte, P., and Savard, G. (2007). An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256.
- Conceição, S. V., da Silva, G. L. C., Lu, D., Nunes, N. T. R., and Pedrosa, G. C. (2015). A demand classification scheme for spare part inventory model subject to stochastic demand and lead time. *Production Planning & Control*, 26(16):1318–1331.
- Dai, H., Khalil, E., Yuyu, Z., Dilkina, B., and Song, L. (2017). Learning Combinatorial Optimization Algorithms over Graphs.
- Dash, G. and Chakraborty, D. (2021). Digital Transformation of Marketing Strategies during a Pandemic: Evidence from an Emerging Economy during COVID-19. *Sustainability*, 13(12):6735.
- De Castro, L., Jos, F., and von Zuben, A. A. (2000). Artificial Immune Systems: Part II - A Survey Of Applications.
- De Castro, L. N. and Von Zuben, F. J. (1999). Artificial immune systems: Part I—basic theory and applications. *Universidade Estadual de Campinas, Dezembro de, Tech. Rep*, 210(1).
- Dehghani, M., Abbasi, B., and Oliveira, F. (2021). Proactive transshipment in the blood supply chain: A stochastic programming approach. *Omega*, 98:102112.
- Dempe, S. and Zemkoho, A. (2020). *Bilevel optimization*. Springer.
- Desaulniers, G., Rakke, J. G., and Coelho, L. C. (2016). A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science*, 50(3):1060–1076.
- Dong, L. and Rudi, N. (2004). Who Benefits from Transshipment? Exogenous vs. Endogenous Wholesale Prices. *Management Science*, 50(5):645–657.

- Dror, M. and Levy, L. (1986). A vehicle routing improvement algorithm comparison of a "greedy" and a matching implementation for inventory routing. *Computers & Operations Research*, 13(1):33–45.
- Duan, L., Zhan, Y., Hu, H., Gong, Y., Wei, J., Zhang, X., and Xu, Y. (2020). Efficiently solving the practical vehicle routing problem: A novel joint learning approach. pages 3054–3063.
- Dudukalov, E. V., Terenina, I. V., Perova, M. V., and Ushakov, D. (2021). Industry 4.0 readiness: the impact of digital transformation on supply chain performance. In *E3S Web of Conferences*, volume 244, page 8020. EDP Sciences.
- Dupacová, J. and Wets, R. (1988). Asymptotic behavior of statistical estimators and of optimal solutions of stochastic optimization problems. *The annals of statistics*, 16(4):1517–1549.
- Ech-Charrat, M. R., Amechnoue, K., and Zouadi, T. (2017). Dynamic Planning of Reusable Containers in a Close-loop Supply Chain under Carbon Emission Constrain. *International Journal of Supply and Operations Management*, 4(4):279–297.
- Ekren, B. Y., Mangla, S. K., Turhanlar, E. E., Kazancoglu, Y., and Li, G. (2021). Lateral inventory share-based models for iot-enabled e-commerce sustainable food supply networks. *Computers & Operations Research*, 130:105237.
- Fahim, P. B. M., Rezaei, J., Montreuil, B., and Tavasszy, L. (2021). Port performance evaluation and selection in the Physical Internet. *Transport Policy*.
- Faruquee, M., Paulraj, A., and Irawan, C. A. (2021). Strategic supplier relationships and supply chain resilience: Is digital transformation that precludes trust beneficial? *International Journal of Operations & Production Management*.
- Fattahi, P. and Tanhatalab, M. (2021). Stochastic inventory-routing problem with lateral transshipment for perishable product. *Journal of Modelling in Management*.
- Fazili, M., Venkatadri, U., Cyrus, P., and Tajbakhsh, M. (2017). Physical internet, conventional and hybrid logistic systems: a routing optimisation-based comparison using the eastern canada road network case study. *International Journal of Production Research*, 55(9):2703–2730.
- Feng, C., Ma, Y., Zhou, G., and Ni, T. (2018). Stackelberg game optimization for integrated production-distribution-construction system in construction supply chain. *Knowledge-Based Systems*, 157:52–67.
- Geiger, M. J. and Sevaux, M. (2011). Practical inventory routing: A problem definition and an optimization method. *arXiv preprint arXiv:1102.5635*.

- Glock, C. H. (2017). Decision support models for managing returnable transport items in supply chains: A systematic literature review. *International Journal of Production Economics*, 183:561–569.
- Gong, G., Deng, Q., Gong, X., Zhang, L., Wang, H., and Xie, H. (2018). A bee evolutionary algorithm for multiobjective vehicle routing problem with simultaneous pickup and delivery. *Mathematical Problems in Engineering*, 2018.
- Govindan, K., Soleimani, H., and Kannan, D. (2015). Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future. *European Journal of Operational Research*, 240(3):603–626.
- Grahovac, J. and Chakravarty, A. (2001). Sharing and lateral transshipment of inventory in a supply chain with expensive low-demand items. *Management Science*, 47(4):579–594.
- Granot, D. and Sošić, G. (2003). A three-stage model for a decentralized distribution system of retailers. *Operations research*, 51(5):771–784.
- Grønhaug, R., Christiansen, M., Desaulniers, G., and Desrosiers, J. (2010). A branch-and-price method for a liquefied natural gas inventory routing problem. *Transportation Science*, 44(3):400–415.
- Gruher, A., Panadero, J., de Armas, J., Moreno Pérez, J. A., and Juan, A. A. (2018). Combining variable neighborhood search with simulation for the inventory routing problem with stochastic demands and stock-outs. *Computers & Industrial Engineering*, 123:278–288.
- GS1 Global Office (2014). Reusable Transport Items within GS1 EANCOM. Technical report, GS1 Global Office, Brussels, Belgium.
- Guimarães, T. A., Schenekemberg, C. M., Coelho, L. C., Scarpin, C. T., and Pécora Jr, J. E. (2020). *Mechanisms for Feasibility and Improvement for Inventory-Routing Problems*.
- Gunawan, A., Vincent, F. Y., Widjaja, A. T., and Vansteenwegen, P. (2019). Simulated Annealing for the Multi-Vehicle Cyclic Inventory Routing Problem. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 691–696. IEEE.
- Han, M., Senellart, P., Bressan, S., and Wu, H. (2016). Routing an autonomous taxi with reinforcement learning. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 2421–2424.
- Hanany, E., Tzur, M., and Levrán, A. (2010). The transshipment fund mechanism: Coordinating the decentralized multilocation transshipment problem. *Naval Research Logistics (NRL)*, 57(4):342–353.

- Hansen, P., Mladenović, N., Brimberg, J., and Pérez, J. A. M. (2019). Variable neighborhood search. In *Handbook of metaheuristics*, pages 57–97. Springer.
- Hassanzadeh Amin, S., Wu, H., and Karaphillis, G. (2018). A perspective on the reverse logistics of plastic pallets in Canada. *Journal of Remanufacturing*, 8(3):153–174.
- Herer, Y. and Tzur, M. (2001). The Dynamic Transshipment Problem. *Naval Research Logistics (NRL)*, 48:386–408.
- Hewitt, M., Nemhauser, G., Savelsbergh, M., and Song, J.-H. (2013). A branch-and-price guided search approach to maritime inventory routing. *Computers & Operations Research*, 40(5):1410–1419.
- Hezarkhani, B. and Kubiak, W. (2010). Transshipment prices and pair-wise stability in coordinating the decentralized transshipment problem. *Behavioral and Quantitative Game Theory: Conference on Future Directions 2010, BQGT 2010*.
- Homem-de Mello, T. and Bayraksan, G. (2014). Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19(1):56–85.
- Hssini, I., Meskens, N., and Riane, F. (2016). Blood products inventory pickup and delivery problem under time windows constraints. In *International Conference on Operations Research and Enterprise Systems*, volume 2, pages 349–356.
- Hu, X., Duenyas, I., and Kapuscinski, R. (2007). Existence of Coordinating Transshipment Prices in a Two-Location Inventory Model. *Management Science*, 53(8):1289–1302.
- Huang, S.-H. and Lin, P.-C. (2010). A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):598–611.
- Huang, X. and Sošić, G. (2010). Transshipment of Inventories: Dual Allocations vs. Transshipment Prices. *Manufacturing & Service Operations Management*, 12(2):299–318.
- Iassinovskaia, G., Limbourg, S., and Riane, F. (2017). The inventory-routing problem of returnable transport items with time windows and simultaneous pickup and delivery in closed-loop supply chains. *International Journal of Production Economics*, 183:570–582.
- IHL, G. (2015). Retail's \$1.1 Trillion Inventory Distortion Problem.
- Ji, S.-f., Peng, X.-s., and Luo, R.-j. (2019). An integrated model for the production-inventory-distribution problem in the physical internet. *International Journal of Production Research*, 57(4):1000–1017.

- Jia, L., Wang, Y., and Lei, F. (2014). Multiobjective bilevel optimization for production-distribution planning problems using hybrid genetic algorithm. *Integrated Computer-Aided Engineering*, 21:77–90.
- Jiang, J., Sun, H., and Zhou, B. (2021). Convergence analysis of sample average approximation for a class of stochastic nonlinear complementarity problems: from two-stage to multistage. *Numerical Algorithms*, pages 1–28.
- Jiang, L. and Anupindi, R. (2010). Customer-driven vs. retailer-driven search: Channel performance and implications. *Manufacturing & Service Operations Management*, 12(1):102–119.
- Johnston, F. R., Boylan, J. E., and Shale, E. A. (2003). An examination of the size of orders from customers, their characterisation and the implications for inventory control of slow moving items. *Journal of the Operational Research Society*, 54(8):833–837.
- Kaboudani, Y., Ghodsypour, S., Kia, H., and Shahmardan, A. (2020). Vehicle routing and scheduling in cross docks with forward and reverse logistics. *Operational Research*, 20.
- Kaempfer, Y. and Wolf, L. (2018). Learning the multiple traveling salesmen problem with permutation invariant pooling networks. *arXiv preprint arXiv:1803.09621*.
- Karakostas, P., Sifaleras, A., and Georgiadis, M. (2020). Variable neighborhood search-based solution methods for the pollution location-inventory-routing problem. *Optimization Letters*.
- Karp, R. (1972). *Reducibility Among Combinatorial Problems*, volume 40.
- Kheirkhah, A. and Rezaei, S. (2015). Using cross-docking operations in a reverse logistics network design: a new approach. *Production Engineering*, 10.
- Kim, T. and Glock, C. (2014). On the use of RFID in the management of reusable containers in closed-loop supply chains under stochastic container return quantities. *Transportation Research Part E Logistics and Transportation Review*, 64:12–27.
- Kleywegt, A. J., Shapiro, A., and Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502.
- Kochańska, E., Łukasik, R. M., and Dzikuć, M. (2021). New circular challenges in the development of take-away food packaging in the covid-19 period. *Energies*, 14(15):4705.

- Kuo, R. and Han, Y. (2011). A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem – a case study on supply chain model. *Applied Mathematical Modelling*, 35(8):3905–3917.
- Kutanoglu, E. and Mahajan, M. (2009). An inventory sharing and allocation method for a multi-location service parts logistics network with time-based service levels. *European Journal of Operational Research*, 194(3):728–742.
- Lacerda, A. P., Xambre, A. R., and Alvelos, H. M. (2016). Applying value stream mapping to eliminate waste: a case study of an original equipment manufacturer for the automotive industry. *International Journal of Production Research*, 54(6):1708–1720.
- Landschützer, C., Ehrentraut, F., and Jodin, D. (2015). Containers for the physical internet: requirements and engineering design related to fmcg logistics. *Logistics Research*, 8(1):8.
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation science*, 43(4):408–416.
- Lee, Y. H., Jung, J., and Lee, K. (2006). Vehicle routing scheduling for cross-docking in supply chain. *Computers & Industrial Engineering*, 51:247–256.
- Lefever, W., Aghezzaf, E.-H., Hadj-Hamou, K., and Penz, B. (2018). Analysis of an improved branch-and-cut formulation for the inventory-routing problem with transshipment. *Computers & Operations Research*, 98:137–148.
- Lengu, D., Syntetos, A. A., and Babai, M. Z. (2014). Spare parts management: Linking distributional assumptions to demand classification. *European Journal of Operational Research*, 235(3):624–635.
- Li, B., Krushinsky, D., Van Woensel, T., and Reijers, H. (2015). An Adaptive Large Neighborhood Search Heuristic for the Share-a-Ride Problem. *Computers & Operations Research*, 66.
- Li, M., Wang, Z., and Chan, F. T. (2016). A robust inventory routing policy under inventory inaccuracy and replenishment lead-time. *Transportation Research Part E: Logistics and Transportation Review*, 91:290–305.
- Li, Y., Guo, H., Wang, L., and Fu, J. (2013). A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under e-supply chain environment. *The Scientific World Journal*, 2013.
- Li, Y., Liao, Y., Hu, X., and Shen, W. (2020). Lateral transshipment with partial request and random switching. *Omega*, 92:102134.
- Li, Z., Xu, J., Shen, W., Lev, B., and Lei, X. (2014). Bi level multi-objective construction site security planning with twofold random phenomenon. *Journal of Industrial and Management Optimization*, 11:595–617.

- Liao, Y., Li, J., Hu, X., Li, Y., and Shen, W. (2020). Application of lateral transshipment in cost reduction of decentralized systems. *Sustainability*, 12:5081.
- Lim, A., Miao, Z., and xu, Z. (2005). Transshipments through crossdocks with inventory and time windows. *Naval Research Logistics - NAV RES LOG*, 52:724–733.
- Limbourg, S., Martin, A., and Paquay, C. (2016). Optimal returnable transport items management.
- Liu, C. (2020). Iterative heuristic for simultaneous allocations of berths, quay cranes, and yards under practical situations. *Transportation Research Part E: Logistics and Transportation Review*, 133(October 2018):1–18.
- Liu, G., Li, L., Chen, J., and Ma, F. (2020). Inventory sharing strategy and optimization for reusable transport items. *International Journal of Production Economics*, 228:107742.
- Lu, H., Zhang, X., and Yang, S. (2019). A learning-based iterative method for solving vehicle routing problems.
- López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., and Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43 – 58.
- Ma, H., Miao, Z., Lim, A., and Rodrigues, B. (2011). Crossdocking distribution networks with setup cost and time window constraint. *Omega*, 39:64–72.
- Ma, Y., Yan, F., Kang, K., and Wei, X. (2016). A novel integrated production-distribution planning model with conflict and coordination in a supply chain network. *Knowledge-Based Systems*, 105:119–133.
- Mangal, D. and Chandna, P. (2010). Lateral transshipment—A technique for inventory control in multi retailer supply chain system. *International Journal of Information Technology and Knowledge Management*, 2(2):311–315.
- Marinakis, Y., Migdalas, A., and Pardalos, P. M. (2007). A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm. *Journal of Global Optimization*, 38(4):555–580.
- Marjani, M., moattar husseini, M., and Karimi, B. (2012). Bi-objective heuristics for multi-item freights distribution planning problem in crossdocking networks. *International Journal of Advanced Manufacturing Technology - INT J ADV MANUF TECHNOL*, 58.
- Meherishi, L., Narayana, S. A., and Ranjani, K. S. (2019). Sustainable packaging for supply chain management in the circular economy: A review. *Journal of Cleaner Production*, 237:117582.

- Meherishi, L., Narayana, S. A., and Ranjani, K. S. (2021). Integrated product and packaging decisions with secondary packaging returns and protective packaging management. *European Journal of Operational Research*, 292(3):930–952.
- Meller, R. D., Montreuil, B., Thivierge, C., and Montreuil, Z. (2012). Functional design of physical internet facilities: A road-based transit center. In *Progress in Material Handling Research*.
- Michel, S. and Vanderbeck, F. (2012). A column-generation based tactical planning method for inventory routing. *Operations research*, 60(2):382–397.
- Minner, S. (2003). Multiple-supplier inventory models in supply chain management: A review. *International Journal of Production Economics*, 81-82:265–279.
- Mirzaei, S. and Seifi, A. (2015). Considering lost sale in inventory routing problems for perishable goods. *Computers & Industrial Engineering*, 87:213–227.
- Mirzapour Al-e-hashem, S. and Rekik, Y. (2014). Multi-product multi-period inventory routing problem with a transshipment option: A green approach. *International Journal of Production Economics*, 157:80 – 88. The International Society for Inventory Research, 2012.
- Mohammed, M. A., Ghani, M. K. A., Hamed, R. I., Mostafa, S. A., Ibrahim, D. A., Jameel, H. K., and Alallah, A. H. (2017). Solving vehicle routing problem by using improved k-nearest neighbor algorithm for best solution. *Journal of Computational Science*, 21:232 – 240.
- Mohtashami, A. (2015). A Novel Dynamic Genetic Algorithm-Based Method for Vehicle Scheduling in Cross Docking Systems with Frequent Unloading Operation. *Computers & Industrial Engineering*, 90.
- Montreuil, B. (2011). Toward a physical internet: meeting the global logistics sustainability grand challenge. *Logistics Research*, 3(2):71–87.
- Montreuil, B., Rougès, J.-F., Cimon, Y., and Poulin, D. (2012). The physical internet and business model innovation. *Technology Innovation Management Review*, 2:32–37.
- Mordor Intelligence (2021). Returnable Transport Packaging (RTP) Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026).
- Moscato, P. et al. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826:1989.
- Mousavi, S. and Vahdani, B. (2017). A robust approach to multiple vehicle location-routing problems with time windows for optimization of cross-docking under uncertainty. *Journal of Intelligent & Fuzzy Systems*, 32:49–62.

- Musa, R., Arnaout, J.-P., and Jung, H. (2010). Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. *Computers & Industrial Engineering*, 59:85–92.
- Myerson, R. B. (1991). *Game Theory*. Harvard University Press.
- Na, B., Sim, M. K., and Lee, W. J. (2019). An Optimal Purchase Decision of Reusable Packaging in the Automotive Industry.
- Nakib, A., Hilia, M., Heliodore, F., and Talbi, E.-G. (2017). Design of metaheuristic based on machine learning: A unified approach. In *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 510–518. IEEE.
- Nananukul, N. (2013). Clustering model and algorithm for production inventory and distribution problem. *Applied Mathematical Modelling*, 37(24):9846–9857.
- Navarro, M., Herath, P., Villarrubia, G., Prieto-Castrillo, F., and Venyagamoorthy, G. (2018). An Evaluation of a Metaheuristic Artificial Immune System for Household Energy Optimization. *Complexity*, 2018:1–11.
- Nia, M. and Alipouri, Y. (2009). *Speeding Up the Genetic Algorithm Convergence Using Sequential Mutation and Circular Gene Methods*.
- OroojlooyJadid, A. and Hajinezhad, D. (2019). A review of cooperative multi-agent deep reinforcement learning. *arXiv preprint arXiv:1908.03963*.
- Pan, S., Nigrelli, M., Ballot, E., Sarraj, R., and Yang, Y. (2014). Perspectives of inventory control models in the physical internet: A simulation study. *Computers & Industrial Engineering*, 84.
- Pan, S., Trentesaux, D., Ballot, E., and Huang, G. Q. (2019). Horizontal collaborative transport: survey of solutions and practical implementation issues. *International Journal of Production Research*, 57(15-16):5340–5361.
- Panigrahi, B. K., Yadav, S. R., Agrawal, S., and Tiwari, M. K. (2007). A clonal algorithm to solve economic load dispatch. *Electric Power Systems Research*, 77(10):1381–1389.
- Panzone, L. A., Larcom, S., and She, P.-W. (2021). Estimating the impact of the covid-19 shock on uk food retailers and the restaurant sector. *Global Food Security*, page 100495.
- Papadimitriou, C. and Steiglitz, K. (1977). On the Complexity of Local Search for the Traveling Salesman Problem. *SIAM J. Comput.*, 6:76–83.
- Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, 58(1):21–51.

- Paterson, C., Kiesmüller, G., Teunter, R., and Glazebrook, K. (2011). Inventory models with lateral transshipments: A review. *European Journal of Operational Research*, 210(2):125–136.
- Peres, I. T., Repolho, H. M., Martinelli, R., and Monteiro, N. J. (2017). Optimization in inventory-routing problem with planned transshipment: A case study in the retail industry. *International Journal of Production Economics*, 193:748–756.
- Pierrard, T. and Coello Coello, C. A. (2012). A Multi-Objective Artificial Immune System Based on Hypervolume BT - Artificial Immune Systems. pages 14–27, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Polat, O. (2017). A parallel variable neighborhood search for the vehicle routing problem with divisible deliveries and pickups. *Computers & Operations Research*, 85:71–86.
- Qu, Y. and Bard, J. F. (2012). A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research*, 39(10):2439–2456.
- Rahim, M. K. I. A., Zhong, Y., Aghezzaf, E.-H., and Aouam, T. (2014). Modelling and solving the multiperiod inventory-routing problem with stochastic stationary demand rates. *International Journal of Production Research*, 52(14):4351–4363.
- Rais, A., Alvelos, F., and Carvalho, M. S. (2014). New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235(3):530–539.
- Reinartz, W., Wiegand, N., and Imschloss, M. (2019). The impact of digital transformation on the retailing value chain. *International Journal of Research in Marketing*, 36(3):350–366.
- Ren, J., Liu, B., and Wang, Z. (2017). An optimization model for multi-type pallet allocation over a pallet pool. *Advances in Mechanical Engineering*, 9(5):1687814017705841.
- Rezaei, S. and Kheirkhah, A. (2017). Applying forward and reverse cross-docking in a multi-product integrated supply chain network. *Production Engineering*, 11:1–15.
- Roch, Y., Ballot, E., and Perraudin, X. (2014). A new framework for the management of returnable “containers” within open supply networks. *Studies in Computational Intelligence*, 594.
- Roldán, R. F., Basagoiti, R., and Coelho, L. C. (2016). Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. *Computers & Operations Research*, 74:14–20.

- Ronen, D. (2002). Marine inventory routing: Shipments planning. *Journal of the Operational Research Society*, 53(1):108–114.
- Rong, Y., Snyder, L. V., and Sun, Y. (2010). Inventory sharing under decentralized preventive transshipments. *Naval Research Logistics (NRL)*, 57(6):540–562.
- Ronzoni, C., Ferrara, A., and Grassi, A. (2015). A stochastic methodology for the optimal management of infrequent demand spare parts in the automotive industry. *IFAC-PapersOnLine*, 48(3):1405–1410.
- Royset, J. O. and Polak, E. (2004). Reliability-based optimal design using sample average approximations. *Probabilistic Engineering Mechanics*, 19(4):331–343.
- Royset, J. O. and Szechtman, R. (2013). Optimal budget allocation for sample average approximation. *Operations Research*, 61(3):762–776.
- Rudi, N., Kapur, S., and Pyke, D. F. (2001). A two-location inventory model with transshipment and local decision making. *Management science*, 47(12):1668–1680.
- Sabba, S. and Chikhi, S. (2012). Integrating the Best 2-Opt Method to Enhance the Genetic Algorithm Execution Time in Solving the Traveler Salesman Problem. *Advances in Intelligent and Soft Computing*, 170:195–208.
- Santoso, T., Ahmed, S., Goetschalckx, M., and Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1):96–115.
- Sarkar, B., Ullah, M., and Kim, N. (2017). Environmental and economic assessment of closed-loop supply chain with remanufacturing and returnable transport items. *Computers & Industrial Engineering*, 111:148–163.
- Savelsbergh, M. and Song, J.-H. (2008). An optimization algorithm for the inventory routing problem with continuous moves. *Computers & Operations research*, 35(7):2266–2282.
- Seyyedabbasi, A., Aliyev, R., Kiani, F., Gulle, M. U., Basyildiz, H., and Shah, M. A. (2021). Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems. *Knowledge-Based Systems*, 223:107044.
- Sgarbossa, F., Peron, M., Lolli, F., and Balugani, E. (2021). Conventional or additive manufacturing for spare parts management: An extensive comparison for Poisson demand. *International Journal of Production Economics*, 233:107993.
- Shao, J., Krishnan, H., and McCormick, S. T. (2011). Incentives for Transshipment in a Supply Chain with Decentralized Retailers. *Manufacturing & Service Operations Management*, 13(3):361–372.

- Shapiro, A., Dentcheva, D., and Ruszczyński, A. (2014). *Lectures on stochastic programming: modeling and theory*. SIAM.
- Shapiro, A. and Philpott, A. (2007). A tutorial on stochastic programming. *Manuscript*. Available at www2.isye.gatech.edu/ashapiro/publications.html, 17.
- Shastri, Y. and Diwekar, U. (2006). An efficient algorithm for large scale stochastic nonlinear programming problems. *Computers & chemical engineering*, 30(5):864–877.
- Shi, C., Lu, J., and Zhang, G. (2005). An extended Kuhn–Tucker approach for linear bilevel programming. *Applied Mathematics and Computation*, 162(1):51–63.
- Shi, X., Liu, W., and Zhang, J. (2021). Present and future trends of supply chain management in the presence of COVID-19: a structured literature review. *International Journal of Logistics Research and Applications*, pages 1–30.
- Silver, D. (2015). *RL Exploration and Exploitation*.
- Singh, S. K., Gupta, Y., Mishra, A., and Darla, S. (2017). Inventory Routing Problem with Simultaneous Pickup and Delivery of Returnable Transport Items with Consideration of Renting and Repairing. *International journal of engineering research and technology*, 6.
- Sinha, A., Malo, P., and Deb, K. (2017). A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295.
- Solyalı, O., Cordeau, J.-F., and Laporte, G. (2012). Robust Inventory Routing Under Demand Uncertainty. *Transportation Science*, 46:327–340.
- Sošić, G. (2006). Transshipment of inventories among retailers: Myopic vs. far-sighted stability. *Management science*, 52(10):1493–1508.
- Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., and van der Vorst, J. G. A. J. (2015). Modeling an Inventory Routing Problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics*, 164:118–133.
- Strack, G., Fortz, B., Riane, F., and Van Vyve, M. (2011). Comparison of heuristic procedures for an integrated model for production and distribution planning in an environment of shared resources. LIDAM Discussion Papers CORE 2011016, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE).

- Stålhane, M., Andersson, H., Christiansen, M., Cordeau, J.-F., and Desaulniers, G. (2012). A branch-price-and-cut method for a ship routing and scheduling problem with split loads. *Computers & Operations Research*, 39:3361–3375.
- Syntetos, A. A., Babai, M. Z., and Altay, N. (2012). On the demand distributions of spare parts. *International Journal of Production Research*, 50(8):2101–2117.
- Tagaras, G. and Vlachos, D. (2002). Effectiveness of stock transshipment under various demand distributions and nonnegligible transshipment times. *Production and Operations Management*, 11(2):183–198.
- Talaei, M., Farhang Moghaddam, B., Pishvaei, M. S., Bozorgi-Amiri, A., and Gholamnejad, S. (2016). A robust fuzzy optimization model for carbon-efficient closed-loop supply chain network design problem: a numerical illustration in electronics industry. *Journal of Cleaner Production*, 113:662–673.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.
- Talbi, E.-G. (2020). Machine learning into metaheuristics: A survey and taxonomy of data-driven metaheuristics. working paper or preprint.
- Tarantilis, C. D., Anagnostopoulou, A. K., and Repoussis, P. P. (2012). Adaptive Path Relinking for Vehicle Routing and Scheduling Problems with Product Returns. *Transportation Science*, 47(3):356–379.
- Tarhini, H., Karam, M., and Jaber, M. Y. (2019). An integrated single-vendor multi-buyer production inventory model with transshipments between buyers. *International Journal of Production Economics*, page 107568.
- Timajchi, A., Mirzapour Al-e-Hashem, S. M., and Rekik, Y. (2019). Inventory routing problem for hazardous and deteriorating items in the presence of accident risk with transshipment option. *International Journal of Production Economics*, 209:302 – 315. The Proceedings of the 19th International Symposium on Inventories.
- Tiwari, M. K., Prakash, Kumar, A., and Mileham, A. R. (2005). Determination of an optimal assembly sequence using the psychoclonal algorithm. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 219(1):137–149.
- Tornese, F., Pazour, J. A., Thorn, B. K., Roy, D., and Carrano, A. L. (2018). Investigating the environmental and economic impact of loading conditions and repositioning strategies for pallet pooling providers. *Journal of Cleaner Production*, 172:155–168.
- Toth, P. and Vigo, D. (2002). *The vehicle routing problem*. SIAM.
- TrackX (2017). A Return on Returnables. Technical report.

- Tran-Dang, H., Krommenacker, N., Charpentier, P., and Kim, D.-S. (2020). Toward the Internet of Things for physical Internet: Perspectives and challenges. *IEEE internet of things journal*, 7(6):4711–4736.
- Turan, B., Minner, S., and Hartl, R. F. (2017). A VNS approach to multi-location inventory redistribution with vehicle routing. *Computers & Operations Research*, 78:526–536.
- Turrini, L. and Meissner, J. (2019). Spare parts inventory management: New evidence from distribution fitting. *European Journal of Operational Research*, 273(1):118–130.
- Twede, D. and Clarke, R. (2005). Supply Chain Issues in Reusable Packaging. *Journal of Marketing Channels*, 12:7–26.
- Ullah, M., Asghar, I., Zahid, M., Omair, M., AlArjani, A., and Sarkar, B. (2021). Ramification of remanufacturing in a sustainable three-echelon closed-loop supply chain management for returnable products. *Journal of Cleaner Production*, 290:125609.
- Upadhyay, R. K. and Kumaraswamidhas, L. A. (2018). Chapter 11 - Bearing failure issues and corrective measures through surface engineering. *Handbook of Materials Failure Analysis*, pages 209–233.
- van Anholt, R. G., Coelho, L. C., Laporte, G., and Vis, I. F. A. (2016). An inventory-routing problem with pickups and deliveries arising in the replenishment of automated teller machines. *Transportation Science*, 50(3):1077–1091.
- Van der Auweraer, S. and Boute, R. (2019). Forecasting spare part demand using service maintenance information. *International Journal of Production Economics*, 213:138–149.
- Van der Heide, G., Buijs, P., Roodbergen, K. J., and Vis, I. F. A. (2018). Dynamic shipments of inventories in shared warehouse and transportation networks. *Transportation Research Part E: Logistics and Transportation Review*, 118:240–257.
- Van Jaarsveld, W. and Dekker, R. (2011). Spare parts stock control for redundant systems using reliability centered maintenance data. *Reliability Engineering & System Safety*, 96(11):1576–1586.
- Verweij, B., Ahmed, S., Kleywegt, A. J., Nemhauser, G., and Shapiro, A. (2003). The sample average approximation method applied to stochastic routing problems: a computational study. *Computational optimization and applications*, 24(2):289–333.
- Viu-Roig, M. and Alvarez-Palau, E. J. (2020). The Impact of E-Commerce-Related Last-Mile Logistics on Cities: A Systematic Literature Review. *Sustainability*, 12(16):1–19.

- Wade, A. C. and Salhi, S. (2002). An investigation into a new class of vehicle routing problem with backhauls. *Omega*, 30(6):479–487.
- Wang, G., Wan, Z., Wang, X., and Lv, Y. (2008). Genetic algorithm based on simplex method for solving linear-quadratic bilevel programming problem. *Computers & Mathematics with Applications*, 56(10):2550–2555.
- Wang, J., Jagannathan, A., Zuo, X., and Murray, C. (2017). Two-layer simulated annealing and tabu search heuristics for a vehicle routing problem with cross docks and split deliveries. *Computers & Industrial Engineering*, 112.
- Wang, Z., Cui, B., Feng, Q., Huang, B., Ren, Y., Sun, B., Yang, D., and Zuo, Z. (2019). An agent-based approach for resources' joint planning in a multi-echelon inventory system considering lateral transshipment. *Comput. Ind. Eng.*, 138:106098.
- Wassan, N., Wassan, N., Nagy, G., and Salhi, S. (2017). The multiple trip vehicle routing problem with backhauls: Formulation and a two-level variable neighbourhood search. *Computers & Operations Research*, 78:454–467.
- White, D. J. and Anandalingam, G. (1993). A penalty function approach for solving bi-level linear programs. *Journal of Global Optimization*, 3(4):397–419.
- Wong, E. Y. C., Lau, H. Y. K., and Mak, K. L. (2010). Immunity-based evolutionary algorithm for optimal global container repositioning in liner shipping. *OR Spectrum*, 32(3):739–763.
- Wong, H., Cattrysse, D., and Van Oudheusden, D. (2005). Stocking decisions for repairable spare parts pooling in a multi-hub system. *International Journal of Production Economics*, 93:309–317.
- Wu, W., Zhou, W., Lin, Y., Xie, Y., and Jin, W. (2021). A hybrid metaheuristic algorithm for location inventory routing problem with time windows and fuel consumption. *Expert Systems with Applications*, 166:114034.
- Xu, D., Li, K., Zou, X., and Liu, L. (2017). An unpaired pickup and delivery vehicle routing problem with multi-visit. *Transportation Research Part E: Logistics and Transportation Review*, 103:218–247.
- Yan, X. and Zhao, H. (2011). Decentralized inventory sharing with asymmetric information. *Operations research*, 59(6):1528–1538.
- Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.
- Yang, Y., Pan, S., and Ballot, E. (2017). Innovative vendor-managed inventory strategy exploiting interconnected logistics services in the physical internet. *International Journal of Production Research*, 55(9):2685–2702.

- Yu, Y., Chu, C., Chen, H., and Chu, F. (2012). Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research*, 197(1):135–158.
- Zhang, D., Liu, Y., M'Hallah, R., and Leung, S. C. H. (2010). A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. *European Journal of Operational Research*, 203(3):550–558.
- Zhang, G., Han, J., and Lu, J. (2016). Fuzzy Bi-level Decision-Making Techniques: A Survey. *International Journal of Computational Intelligence Systems*, 9(sup1):25–34.
- Zhang, Q., Segerstedt, A., Tsao, Y.-C., and Liu, B. (2015). Returnable packaging management in automotive parts logistics: Dedicated mode and shared mode. *International Journal of Production Economics*, 168:234–244.
- Zhao, F., Wu, D., Liang, L., and Dolgui, A. (2016). Lateral inventory transshipment problem in online-to-offline supply chain. *International Journal of Production Research*, 54(7):1951–1963.
- Zhao, H. and Bisi, A. (2010). Optimal operating policies in a commodity trading market with the manufacturer's presence. *Naval Research Logistics (NRL)*, 57(2):127–148.
- Zhao, H., Deshpande, V., and Ryan, J. K. (2005). Inventory Sharing and Rationing in Decentralized Dealer Networks. *Management Science*, 51(4):531–547.
- Zhao, Q.-H., Chen, S., and Zang, C.-X. (2008). Model and algorithm for inventory/routing decision in a three-echelon logistics system. *European Journal of Operational Research*, 191(3):623–635.
- Zhou, K. and Song, R. (2019). Location Model of Pallet Service Centers Based on the Pallet Pool Mode. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 1185–1189.
- Zuluaga, J., Thiell, M., and Perales, R. (2016). Reverse cross-docking. *Omega*, 66.

Appendices

A - Computational results for the classical IRP (Chapter 2)

Tables A.1, A.2, A.3, A.4, A.5 and A.6 provide computational results for IRP on large instances. For each algorithm B-C and I-B-C, the tables report the best solutions found so far (noted $Z1$ and $Z2$ respectively) and the corresponding CPU in second. For the matheuristic, tables provide the worst solution (WS), the best solution (BS), the average of the best solutions found (\overline{BFS}), and the average of the scaled CPU ($\overline{S_CPU}$) with regards to hardware performances (for a fair comparison the actual CPU was limited so as its corresponding S-CPU would be comparable to the limit time of I-B-C: 7200s). Tables also provide the contribution of the constructive phase (CPC in %) in improving the quality of the final solution. Finally, for each $\overline{S_CPU}$, tables report the computational time needed in the constructive phase (CP) and improvement phase (IP).

Table A.1: Computational results for large scale benchmark instances using B-C, I-B-C and matheuristic (k=1, high cost)

Instances	B-C		I-B-C		Matheuristic						
	Z1	CPU	Z2	CPU	WS	BS	BSF	CPC (%)	S CPU		
									CP	IP	Total
1P50N1K6T	30,189	3,056	30,189	744	30,552	30,190	30,392	48	1314	1458	2772
1P50N1K6T	29,790	3,334	29,790	730	30,088	29,790	29,900	44	1221	1206	2427
1P50N1K6T	29,791	4,020	29,791	1,399	30,089	29,791	29,801	45	1902	2191	4093
1P50N1K6T	31,518	5,737	31,518	1,910	31,897	31,519	31,819	50	1894	2330	4224
1P50N1K6T	29,240	684	29,240	3,586	29,563	29,241	29,269	36	1110	1281	2391
1P50N1K6T	31,903	28,320	31,903	3,477	32,223	31,904	31,934	30	1048	1082	2130
1P50N1K6T	29,735	13,561	29,734	1,412	30,062	29,735	29,892	44	1990	1945	3935
1P50N1K6T	25,954	21,552	25,954	1,312	26,241	25,955	26,228	43	779	1020	1799
1P50N1K6T	30,193	20,581	30,193	1,401	30,495	30,193	30,494	44	1051	1368	2419
1P50N1K6T	31,338	1,879	31,338	2,295	31,652	31,339	31,456	41	599	652	1250
1P100N1K6T	57,459	86,400	57,334	7,200	58,359	57,716	57,970	31	1641	2145	3786
1P100N1K6T	53,510	86,400	53,311	7,200	53,969	53,329	53,703	36	1030	1194	2224
1P100N1K6T	58,505	86,400	58,421	7,200	59,195	58,493	59,109	44	2028	2138	4166
1P100N1K6T	51,554	86,400	51,552	7,200	52,121	51,554	51,755	34	651	749	1400
1P100N1K6T	57,977	86,400	57,943	4,705	58,537	57,957	58,156	32	610	681	1291
1P100N1K6T	55,088	86,400	55,091	7,200	55,695	55,089	55,473	50	1630	1662	3292
1P100N1K6T	56,077	86,400	56,054	7,200	56,620	56,059	56,422	35	863	842	1705
1P100N1K6T	56,057	86,400	55,052	7,200	56,193	55,637	55,722	50	1016	1085	2101
1P100N1K6T	59,426	86,400	58,483	7,200	60,074	59,362	59,694	47	1695	1800	3495
1P100N1K6T	56,588	86,400	56,354	7,200	57,053	56,488	56,489	48	1788	2087	3875
1P200N1K6T	136,337	86,400	111,200	7,200	117,571	116,177	117,063	42	974	1267	2241
1P200N1K6T	141,543	86,400	112,350	7,200	118,102	116,817	117,974	50	1875	2260	4135
1P200N1K6T	123,147	86,400	108,335	7,200	113,693	112,456	112,785	35	1758	2322	4080
1P200N1K6T	129,615	86,400	109,413	7,200	117,458	116,295	117,426	32	1003	1127	2130
1P200N1K6T	126,552	86,400	109,376	7,200	113,762	109,106	113,328	39	1371	1752	3123
1P200N1K6T	136,513	86,400	109,403	7,200	111,321	110,219	110,843	52	1923	1860	3783
1P200N1K6T	111,186	86,400	97,939	7,200	105,586	104,334	105,574	33	841	1065	1906
1P200N1K6T	115,946	86,400	101,754	7,200	106,017	104,967	105,701	35	634	700	1334
1P200N1K6T	136,819	54,474	104,847	7,200	106,143	104,988	105,623	42	857	923	1780
1P200N1K6T	142,796	86,400	109,056	7,200	114,987	113,736	114,858	40	727	740	1467
Average	72078.2	59959.9	64430.7	5325.7	66510.6	65795.4	66228.4	41.1	1260.8	1431.0	2691.8

* All values are rounded up for the sake of a better presentation

Table A.2: Computational results for large scale benchmark instances using B-C, I-B-C and matheuristic (k=1, low cost)

Instances	B-C		I-B-C		Matheuristic						
	Z1	CPU	Z2	CPU	WS	BS	BSF	CPC (%)	S CPU		
									CP	IP	Total
1P50N1K6T	9,976	86,400	9,966	1,485	10,085	9,975	9,987	50	986	974	1960
1P50N1K6T	10,632	2,536	10,632	758	10,749	10,632	10,678	51	631	827	1458
1P50N1K6T	10,511	1,355	10,511	4,575	10,627	10,511	10,569	40	656	659	1315
1P50N1K6T	10,513	60,289	10,513	2,181	10,630	10,514	10,520	31	1101	1449	2550
1P50N1K6T	10,113	2,416	10,113	5,349	10,214	10,113	10,171	46	1150	1450	2600
1P50N1K6T	10,148	86,400	10,148	3,831	10,757	10,629	10,690	45	1294	1404	2698
1P50N1K6T	9,982	14,698	9,982	1,331	10,103	9,983	10,009	36	1385	1620	3005
1P50N1K6T	10,299	86,400	10,299	4,795	10,413	10,300	10,327	43	1023	1341	2364
1P50N1K6T	10,010	6,326	10,010	1,731	10,130	10,010	10,086	51	973	1279	2252
1P50N1K6T	9,659	3,523	9,659	3,912	9,766	9,660	9,731	38	1194	1239	2433
1P100N1K6T	15,649	86,400	15,639	7,200	15,828	15,640	15,745	31	1092	1279	2371
1P100N1K6T	14,697	86,400	14,551	7,200	14,756	14,581	14,678	52	1066	1406	2472
1P100N1K6T	16,155	86,400	15,539	7,200	16,089	15,898	16,021	40	1076	1273	2349
1P100N1K6T	14,644	86,400	14,643	7,200	14,804	14,643	14,669	48	1232	1252	2484
1P100N1K6T	15,235	86,400	15,222	7,200	15,410	15,227	15,325	50	1511	1669	3180
1P100N1K6T	15,769	86,400	15,226	7,200	15,427	15,259	15,344	51	1193	1218	2411
1P100N1K6T	15,538	86,400	15,321	7,200	15,665	15,495	15,620	48	815	1022	1837
1P100N1K6T	15,279	86,400	15,041	7,200	15,383	15,201	15,207	35	1422	1607	3029
1P100N1K6T	17,190	86,400	15,563	7,200	15,999	15,825	15,832	46	1536	1598	3134
1P100N1K6T	16,145	86,400	15,464	7,200	16,233	16,056	16,223	39	1494	1485	2979
1P200N1K6T	32,683	86,400	24,373	7,200	27,225	26,955	27,168	42	1241	1407	2648
1P200N1K6T	34,033	86,400	24,708	7,200	25,503	25,226	25,427	51	1292	1538	2830
1P200N1K6T	33,317	86,400	23,914	7,200	24,888	24,593	24,593	43	1143	1343	2486
1P200N1K6T	34,004	86,400	24,396	7,200	25,025	24,777	24,826	38	965	1118	2083
1P200N1K6T	35,487	86,400	25,216	7,200	26,250	25,990	26,024	36	841	952	1793
1P200N1K6T	33,360	86,400	23,917	7,200	24,991	24,695	24,751	52	820	836	1656
1P200N1K6T	32,774	86,400	23,977	7,200	24,281	24,017	24,035	48	586	714	1300
1P200N1K6T	33,489	79,789	23,165	7,200	24,685	24,392	24,615	37	1179	1227	2406
1P200N1K6T	35,173	86,400	24,729	7,200	25,034	24,737	24,924	50	806	812	1618
1P200N1K6T	34,872	86,400	23,597	7,200	25,088	24,840	24,897	34	871	871	1742
Average	19,911.2	69057.7	16,534.5	5798.3	17,067.9	16,879.1	16,956.4	43.3	1085.8	1229.0	2314.8

* All values are rounded up for the sake of a better presentation

Table A.3: Computational results for large scale benchmark instances using B-C, I-B-C and matheuristic (k=2, high cost)

Instances	B-C		I-B-C		Matheuristic						
	Z1	CPU	Z2	CPU	WS	BS	BSF	CPC (%)	S CPU		
									CP	IP	Total
1P50N2K6T	31,783	86,400	31,557	7200	31,979	31,600	31,819	36	1715	1755	3470
1P50N2K6T	31,121	86,400	30,957	7200	31,429	31,118	31,359	41	1435	1459	2894
1P50N2K6T	30,439	86,400	30,377	7200	30,755	30,420	30,513	40	1008	1225	2233
1P50N2K6T	32,722	86,400	32,288	7200	32,977	32,586	32,627	48	1957	2341	4298
1P50N2K6T	30,115	86,400	30,100	7200	30,441	30,110	30,278	37	1523	1932	3455
1P50N2K6T	33,495	86,400	33,182	7200	33,602	33,269	33,485	38	1219	1544	2763
1P50N2K6T	31,840	86,400	31,315	7200	31,888	31,541	31,773	33	1701	1723	3424
1P50N2K6T	28,552	86,400	27,709	7200	28,381	28,072	28,252	41	1106	1305	2411
1P50NK6T	31,415	86,400	31,287	7200	31,723	31,409	31,605	51	1159	1407	2566
1P50N2K6T	32,788	86,400	32,691	7200	33,027	32,700	33,000	49	1575	1680	3255
1P100N2K6T	91,823	86,400	58,447	7200	61,879	61,266	61,934	41	1972	2245	4217
1P100N2K6T	59,861	86,400	54,791	7200	56,658	56,097	56,181	43	1597	1741	3338
1P100N2K6T	114,040	86,400	59,528	7200	62,469	61,789	62,456	46	1117	1179	2296
1P100N2K6T	55,344	86,400	52,743	7200	54,607	54,066	54,407	37	1601	1628	3229
1P100N2K6T	67,259	86,400	60,179	7200	65,081	64,437	64,833	45	1185	1154	2339
1P100N2K6T	93,171	86,400	57,456	7200	61,983	61,369	61,490	36	2071	2002	4073
1P100N2K6T	111,776	86,400	57,124	7200	62,944	62,198	62,590	52	1850	2415	4265
1P100N2K6T	142,585	86,400	57,247	7200	61,402	60,734	61,274	30	1309	1731	3040
1P100N2K6T	70,293	86,400	60,189	7200	65,692	65,042	65,214	37	1038	1337	2375
1P100N2K6T	89,315	86,400	58,088	7200	63,037	62,290	62,368	52	1979	2277	4256
1P200N2K6T	-	-	114,641	7200	116,401	115,021	115,109	40	1074	1206	2280
1P200N2K6T	-	-	116,632	7200	119,203	118,023	118,736	52	1276	1593	2869
1P200N2K6T	-	-	112,425	7200	114,703	113,567	114,139	30	1480	1770	3250
1P200N2K6T	-	-	112,426	7200	127,441	125,930	127,035	37	1251	1427	2678
1P200N2K6T	-	-	113,247	7200	118,170	116,768	116,875	36	1555	1678	3233
1P200N2K6T	-	-	113,719	7200	123,944	122,717	123,466	31	1444	1588	3032
1P200N2K6T	-	-	103,707	7200	112,197	111,086	111,532	40	1468	1560	3028
1P200N2K6T	-	-	108,320	7200	111,158	110,057	110,266	42	1925	1974	3899
1P200N2K6T	-	-	109,730	7200	118,334	117,162	118,250	35	1247	1505	2752
1P200N2K6T	-	-	114,489	7200	119,217	117,920	118,889	39	1585	1633	3218
Average	60,486.7	86400.0	66,886.3	7200.0	70,424.0	69,678.8	70,058.5	40.4	1481	1667	3148

* All values are rounded up for the sake of a better presentation

Table A.4: Computational results for large scale benchmark instances using B-C, I-B-C and matheuristic (k=2, low cost)

Instances	B-C		I-B-C		Matheuristic						
	Z1	CPU	Z2	CPU	WS	BS	BSF	CPC (%)	S CPU		
									CP	IP	Total
1P50N2K6T	12209	86400	11317	7200	11,615	11,500	11,542	42	1114	1246	2360
1P50N2K6T	11820	86400	11705	7200	11,850	11,709	11,844	45	1467	1683	3150
1P50N2K6T	11238	86400	11041	7200	11,301	11,189	11,221	37	1064	1209	2273
1P50N2K6T	11740	86400	11152	7200	11,286	11,174	11,217	37	1264	1310	2574
1P50N2K6T	10856	86400	10873	7200	10,986	10,866	10,868	40	1167	1199	2366
1P50N2K6T	11662	86400	11381	7200	11,620	11,494	11,500	45	957	1248	2205
1P50N2K6T	12012	86400	11530	7200	11,765	11,637	11,731	42	1195	1319	2514
1P50N2K6T	13379	86400	12130	7200	13,362	13,217	13,353	45	1498	1728	3226
1P50N2K6T	11302	86400	11079	7200	11,294	11,171	11,231	45	1145	1416	2561
1P50N2K6T	11475	86400	10891	7200	11,482	11,368	11,381	41	1263	1379	2642
1P100N2K6T	50284	86400	17082	7200	18,481	18,280	18,434	46	1671	1624	3295
1P100N2K6T	72767	86400	16000	7200	17,370	17,164	17,220	37	1078	1185	2263
1P100N2K6T	20837	86400	17145	7200	17,494	17,287	17,409	50	1235	1347	2582
1P100N2K6T	42634	86400	15423	7200	16,406	16,244	16,298	46	1368	1460	2828
1P100N2K6T	72016	86400	16449	7200	17,677	17,502	17,506	48	1097	1316	2413
1P100N2K6T	52945	86400	17220	7200	18,727	18,505	18,520	43	1637	1604	3241
1P100N2K6T	47628	86400	16780	7200	17,121	16,935	16,942	32	1222	1480	2702
1P100N2K6T	47360	86400	17773	7200	18,991	18,784	18,939	46	1323	1340	2663
1P100N2K6T	48667	86400	16979	7200	17,519	17,346	17,406	35	1610	1617	3227
1P100N2K6T	48398	86400	16259	7200	17,576	17,385	17,547	50	1427	1755	3182
1P200N2K6T	-	-	28662	7200	30,794	30,459	30,786	34	1012	1323	2335
1P200N2K6T	-	-	27535	7200	29,984	29,658	29,914	48	1400	1382	2782
1P200N2K6T	-	-	27828	7200	29,402	29,082	29,150	43	1394	1603	2997
1P200N2K6T	-	-	28333	7200	30,628	30,295	30,527	46	1652	1608	3260
1P200N2K6T	-	-	29083	7200	30,530	30,168	30,195	37	1154	1207	2361
1P200N2K6T	-	-	28551	7200	31,345	30,973	31,006	34	1275	1408	2683
1P200N2K6T	-	-	28293	7200	29,652	29,301	29,361	46	1118	1221	2339
1P200N2K6T	-	-	29290	7200	31,366	30,994	31,248	33	1289	1609	2898
1P200N2K6T	-	-	29077	7200	30,668	30,364	30,446	32	1337	1495	2832
1P200N2K6T	-	-	27265	7200	28,913	28,626	28,807	48	1460	1731	3191
Average	31,061.3	86400.0	18,804.2	7200.0	19,906.8	19,689.2	19,785.0	41.8	1296.4	1435.1	2731.5

* All values are rounded up for the sake of a better presentation

Table A.5: Computational results for large scale benchmark instances using B-C, I-B-C and matheuristic (k=3, high cost)

Instances	B-C		I-B-C		Matheuristic						
	Z1	CPU	Z2	CPU	WS	BS	BSF	CPC (%)	CP	S_CPU IP	Total
1P50N3K6T	36,497	86,400	33,449	7200	34,225	33,819	34,189	49	1926	1917	3843
1P50N3K6T	37,328	86,400	33,594	7200	34,477	34,068	34,091	31	2048	2250	4298
1P50N3K6T	33,124	86,400	32,148	7200	32,748	32,424	32,480	49	1978	2581	4559
1P50N3K6T	41,325	86,400	34,533	7200	35,751	35,397	35,568	40	2168	2579	4747
1P50N3K6T	34,070	86,400	32,067	7200	32,737	32,381	32,454	30	2183	2107	4290
1P50N3K6T	38,640	86,400	35,892	7200	36,491	36,058	36,175	34	1839	1857	3696
1P50N3K6T	35,795	86,400	33,496	7200	34,768	34,356	34,594	39	1663	1895	3558
1P50N3K6T	33,535	86,400	30,535	7200	32,161	31,843	32,110	46	2064	2273	4337
1P50N3K6T	36,437	86,400	33,529	7200	33,919	33,583	33,849	33	1849	2306	4155
1P50N3K6T	36,330	86,400	35,027	7200	35,550	35,163	35,491	41	1487	1904	3391
1P100N3K6T	110,146	86,400	62,597	7200	66,021	65,367	65,441	42	1561	1963	3524
1P100N3K6T	107,683	86,400	60,160	7200	65,560	64,847	65,221	42	1666	1682	3348
1P100N3K6T	165,084	86,400	64,933	7200	68,012	67,272	67,837	50	1705	2220	3925
1P100N3K6T	168,527	86,400	54,369	7200	58,847	58,207	58,833	39	1852	2321	4173
1P100N3K6T	167,986	86,400	63,306	7200	70,618	69,919	70,139	46	2239	2316	4555
1P100N3K6T	109,981	86,400	60,243	7200	63,547	62,793	62,978	38	1818	2348	4166
1P100N3K6T	107,437	86,400	61,319	7200	66,154	65,499	65,583	45	2300	2253	4553
1P100N3K6T	106,218	86,400	61,236	7200	66,736	65,945	66,294	44	1858	2153	4011
1P100N3K6T	184,721	86,400	64,627	7200	68,230	67,421	67,926	44	1820	1825	3645
1P100N3K6T	108,231	86,400	60,377	7200	63,560	62,931	63,070	48	1745	1710	3455
1P200N3K6T	-	-	120,512	7200	124,770	123,412	123,770	42	2027	1991	4018
1P200N3K6T	-	-	122,979	7200	129,930	128,644	129,030	51	1647	1978	3625
1P200N3K6T	-	-	118,259	7200	125,101	123,618	124,148	46	2157	2371	4528
1P200N3K6T	-	-	117,705	7200	120,322	119,131	119,897	50	1728	1800	3528
1P200N3K6T	-	-	118,737	7200	123,205	121,864	121,916	41	1738	1882	3620
1P200N3K6T	-	-	118,640	7200	123,679	122,333	123,539	43	1858	2240	4098
1P200N3K6T	-	-	107,266	7200	115,668	114,297	115,369	44	1593	2106	3699
1P200N3K6T	-	-	112,216	7200	116,104	114,840	116,026	31	1654	2166	3820
1P200N3K6T	-	-	113,994	7200	118,338	117,167	117,777	44	1963	2377	4340
1P200N3K6T	-	-	117,407	7200	128,272	126,876	127,240	32	1849	2081	3930
Average	84,954.7	86400.0	70,505.0	7200.0	74,183.4	73,382.5	73,767.8	41.8	1866.2	2115.0	3981.2

* All values are rounded up for the sake of a better presentation

Table A.6: Computational results for large scale benchmark instances using B-C, I-B-C and matheuristic (k=3, low cost)

Instances	B-C		I-B-C		Matheuristic						
	Z1	CPU	Z2	CPU	WS	BS	BSF	CPC (%)	S CPU		
									CP	IP	Total
1P50N3K6T	15,533	86,400	13,337	7200	14,396	14,225	14,254	49	1509	1643	3152
1P50N3K6T	17,992	86,400	14,112	7200	14,758	14,597	14,599	43	1681	2050	3731
1P50N3K6T	15,425	86,400	12,400	7200	13,411	13,278	13,311	40	1500	1939	3439
1P50N3K6T	17,315	86,400	13,606	7200	14,186	14,032	14,170	46	1618	1869	3487
1P50N3K6T	15,651	86,400	12,938	7200	14,084	13,917	13,947	47	1622	1865	3487
1P50N3K6T	16,468	86,400	13,720	7200	13,971	13,833	13,898	51	1827	1760	3587
1P50N3K6T	17,899	86,400	13,940	7200	14,633	14,459	14,506	51	1541	1500	3041
1P50N3K6T	19,411	86,400	14,866	7200	15,273	15,107	15,116	46	1579	1551	3130
1P50N3K6T	17,759	86,400	13,467	7200	14,717	14,571	14,680	50	1525	1832	3357
1P50N3K6T	16,003	86,400	12,664	7200	12,985	12,844	12,971	43	1387	1757	3144
1P100N3K6T	70,571	86,400	20,381	7200	22,418	22,152	22,367	30	1610	1745	3355
1P100N3K6T	68,421	86,400	17,711	7200	19,807	19,572	19,663	39	1622	2119	3741
1P100N3K6T	64,860	86,400	20,559	7200	21,487	21,253	21,393	47	1607	1704	3311
1P100N3K6T	66,067	86,400	18,005	7200	19,850	19,653	19,739	41	1797	1768	3565
1P100N3K6T	66,508	86,400	20,571	7200	22,825	22,599	22,690	40	1635	2028	3663
1P100N3K6T	69,777	86,400	21,411	7200	22,258	22,038	22,097	42	1433	1894	3327
1P100N3K6T	66,054	86,400	20,269	7200	22,282	22,040	22,127	37	1780	1855	3635
1P100N3K6T	66,338	86,400	21,003	7200	23,030	22,757	22,824	45	1494	1731	3225
1P100N3K6T	71,530	86,400	21,434	7200	22,417	22,195	22,383	46	1447	1676	3123
1P100N3K6T	67,377	86,400	20,169	7200	22,222	22,002	22,170	35	1561	1784	3345
1P200N3K6T	-	-	32,591	7200	33,472	33,140	33,230	37	1456	1813	3269
1P200N3K6T	-	-	33,811	7200	34,792	34,414	34,658	46	1526	1600	3126
1P200N3K6T	-	-	32,713	7200	32,421	32,037	32,242	47	1855	1920	3775
1P200N3K6T	-	-	32,429	7200	34,271	33,898	34,009	33	1563	2002	3565
1P200N3K6T	-	-	33,703	7200	33,766	33,432	33,719	50	1616	1821	3437
1P200N3K6T	-	-	32,847	7200	34,020	33,683	33,794	48	1531	1707	3238
1P200N3K6T	-	-	33,191	7200	34,356	33,983	34,211	39	1590	1578	3168
1P200N3K6T	-	-	32,764	7200	33,549	33,184	33,489	36	1755	1705	3460
1P200N3K6T	-	-	33,670	7200	33,825	33,424	33,600	34	1500	1873	3373
1P200N3K6T	-	-	32,769	7200	33,937	33,534	33,662	35	1637	2133	3770
Average	42,347.9	86400.0	22,235.0	7200.0	23,314.0	23,061.8	23,184.0	42.5	1593.5	1807.4	3400.9

* All values are rounded up for the sake of a better presentation

B - Computational results for different models and demand distributions under consideration (Chapter 3)

Tables B.1, B.3, B.5, B.7 and B.9 report costs computed for the first and second stage (FSC, SSC), standard deviation regarding the upper and lower bound (UB, LB), and CPU time in second. Tables B.2, B.4, B.6, B.8 and B.10 provide for all instances under consideration the breakdown of costs namely: Transportation (T), Inventory (I), Lost sales (LS), substitution (S) and transshipment (Ts). All experiments are performed for four different models: SIRP, SIRP with Transshipment (SIRPT), SIRP with Substitution (SIRPS) and finally SIRPTS. They also report cost saving (SV) computed with regard to total cost.

Table B.1: Computational results for a number of product varying between 20 and 40 - number of customers equal to 10

220

Instances	Model	PD					SPD					NBD				
		FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)
n10m20k3T3	SIRP	2798.9	3847.7	0.63	0.37	1128	3666.0	4370.0	0.27	0.60	1167	3695.6	7296.6	0.63	0.47	1118
	SIRPT	2062.1	3410.5	0.76	0.50	903	3022.8	3588.7	0.32	0.78	802	3053.9	6253.9	0.83	0.77	1099
	SIRPS	2227.2	3548.8	0.54	0.66	1125	2524.1	4104.4	0.83	0.84	992	3047.7	5781.1	0.78	0.78	844
	SIRPTS	1758.3	2750.2	0.44	0.33	857	2003.1	2831.9	0.70	0.71	1025	2527.9	4311.6	0.90	0.88	1043
n10m20k3T5	SIRP	4870.5	9247.0	0.39	0.79	1148	5888.8	9106.9	0.22	0.70	814	6612.1	11648.2	0.82	0.66	1191
	SIRPT	4672.4	8111.3	0.66	0.43	1107	6177.2	6837.8	0.67	0.34	962	6054.8	9824.5	0.44	0.69	907
	SIRPS	4964.4	7269.2	0.34	0.54	868	5533.4	7868.9	0.28	0.52	1137	6145.8	8749.4	0.85	0.86	1056
	SIRPTS	4176.7	5929.0	0.49	0.43	864	5480.6	6978.1	0.36	0.38	1151	4895.5	7869.9	0.80	0.63	813
n10m20k3T7	SIRP	8088.0	13555.1	0.75	0.49	1048	10820.1	14337.1	0.75	0.29	1078	10218.2	20347.8	0.34	0.53	838
	SIRPT	6385.2	12074.5	0.40	0.61	878	7589.9	12273.7	0.77	0.59	1140	8347.3	16319.7	0.42	0.46	1077
	SIRPS	6810.9	10353.7	0.53	0.78	1115	8856.4	10447.0	0.71	0.88	1014	7459.2	13505.3	0.86	0.51	1143
	SIRPTS	7067.9	7483.0	0.55	0.53	1063	8226.6	9571.5	0.69	0.61	865	10104.9	7933.0	0.96	0.63	959
n10m30k3T3	SIRP	3084.7	5039.4	0.74	0.68	1019	3652.6	5924.3	0.44	0.43	837	4148.2	6388.2	0.58	0.84	1055
	SIRPT	2427.5	4395.2	0.70	0.66	910	2807.6	4683.3	0.22	0.28	906	3686.1	5437.9	0.41	0.36	959
	SIRPS	2436.1	4574.2	0.34	0.41	986	3886.3	3442.3	0.48	0.47	982	3658.3	5247.0	0.54	0.40	1105
	SIRPTS	2517.8	2584.6	0.49	0.53	934	3840.3	1905.5	0.55	0.41	1024	3447.7	3542.7	0.56	0.60	989
n10m30k3T5	SIRP	6083.3	8859.7	0.71	0.36	939	8542.0	12100.8	0.58	0.87	990	8419.3	16112.4	0.77	0.47	978
	SIRPT	5450.8	8961.7	0.41	0.71	1088	7313.1	10153.2	0.48	0.68	1167	6252.1	12762.7	0.90	0.52	1122
	SIRPS	4988.7	8626.8	0.49	0.62	934	5867.1	8881.7	0.72	0.79	1199	7295.5	11908.3	0.64	0.36	1111
	SIRPTS	6067.4	5052.7	0.49	0.53	805	6743.4	9266.6	0.70	0.67	881	6615.5	9690.9	0.73	0.84	900
n10m30k3T7	SIRP	8798.7	12306.3	0.47	0.41	900	10931.3	15491.9	0.33	0.58	1140	11103.5	19367.1	0.95	0.55	1129
	SIRPT	7932.9	10834.3	0.35	0.37	829	9578.6	12985.8	0.58	0.33	876	9564.3	18109.8	0.38	0.55	953
	SIRPS	7455.6	11160.1	0.65	0.69	808	8764.8	12597.2	0.75	0.34	1014	9748.9	17789.0	0.86	0.80	1160
	SIRPTS	7834.8	8698.5	0.34	0.53	991	10454.3	8178.8	0.22	0.51	954	9244.8	15013.5	0.53	0.45	949
n10m40k3T3	SIRP	3473.6	6124.7	0.69	0.44	864	5565.5	7374.5	0.26	0.55	1032	3971.1	11407.6	0.97	0.77	812
	SIRPT	3364.6	5164.2	0.42	0.60	1191	4504.4	6276.8	0.41	0.30	1151	5199.9	8136.6	0.38	0.94	1194
	SIRPS	2880.7	5394.8	0.73	0.37	850	4368.9	6194.1	0.51	0.50	1038	3925.1	6697.6	0.53	0.41	971
	SIRPTS	3765.6	3010.0	0.60	0.44	1191	4803.6	5964.5	0.80	0.23	841	5337.6	4413.5	0.91	0.53	1063
n10m40k3T5	SIRP	8881.1	12360.5	0.74	0.37	830	8084.9	13411.8	0.21	0.21	1154	8370.2	16800.2	0.48	0.36	1172
	SIRPT	7635.5	10458.2	0.63	0.45	909	7827.3	12912.7	0.39	0.68	892	8831.9	14358.9	0.67	0.94	1161
	SIRPS	6751.0	11775.4	0.44	0.74	893	8893.0	11758.6	0.66	0.69	894	10033.7	13440.4	0.79	0.74	895
	SIRPTS	6209.1	11597.3	0.49	0.47	1114	8662.1	9869.4	0.47	0.36	1115	8085.0	13356.6	0.82	0.88	1096

n10m40k3T7	SIRP	8772.5	15911.0	0.66	0.46	980	11418.3	15469.0	0.28	0.72	911	12939.7	21166.0	0.33	0.44	877
	SIRPT	8401.5	13118.7	0.47	0.38	832	10141.3	14782.3	0.30	0.77	892	10757.3	16359.6	0.80	0.70	843
	SIRPS	7626.9	13291.7	0.62	0.56	856	8672.7	15714.3	0.85	0.31	830	9820.5	17386.1	0.72	0.88	836
	SIRPTS	8290.9	10418.8	0.46	0.60	1093	11441.5	10550.0	0.22	0.68	1111	9247.3	16225.4	0.71	0.96	1037

Table B.2: Breakdown of cost for a number of product varying between 20 and 40 - number of customers equal to 10

Instances	Model	PD						SPD						NBD					
		T	I	LS	S	Ts	SV (%)	T	I	LS	S	Ts	SV (%)	T	I	LS	S	Ts	SV (%)
n10m20k3T3	SIRP	2799	3016	832	0	0		3666	3016	1354	0	0		3696	6006	1291	0	0	
	SIRPT	2062	1543	533	0	1334	18	3023	1543	744	0	1301	18	3054	2517	1050	0	2687	15
	SIRPS	2227	1836	614	1099	0	13	2524	1836	749	1519	0	18	3048	2899	1030	1852	0	20
	SIRPTS	1758	423	315	1003	1010	32	2003	423	253	1068	1087	40	2528	465	171	1796	1879	38
n10m20k3T5	SIRP	4871	5818	3429	0	0		5889	5818	3288	0	0		6612	7950	3698	0	0	
	SIRPT	4672	3017	2580	0	2514	9	6177	3017	1513	0	2308	13	6055	4966	1291	0	3568	13
	SIRPS	4964	3625	1210	2434	0	13	5533	3625	1362	2882	0	11	6146	4214	1713	2823	0	18
	SIRPTS	4177	1151	1007	1967	1804	28	5481	1151	456	2515	2857	17	4896	1016	493	3369	2992	30
n10m20k3T7	SIRP	8088	9040	4516	0	0		10820	9040	5298	0	0		10218	13865	6483	0	0	
	SIRPT	6385	6426	1873	0	3776	15	7590	6426	1835	0	4014	21	8347	6318	2861	0	7140	19
	SIRPS	6811	5861	1031	3462	0	21	8856	5861	1328	3258	0	23	7459	5761	2584	5160	0	31
	SIRPTS	7068	1812	584	2601	2486	33	8227	1812	969	3541	3250	29	10105	1547	594	2540	3252	41
n10m30k3T3	SIRP	3085	4434	605	0	0		3653	4434	1490	0	0		4148	4914	1474	0	0	
	SIRPT	2428	2345	478	0	1572	16	2808	2345	595	0	1743	22	3686	2734	518	0	2186	13
	SIRPS	2436	1822	1009	1742	0	14	3886	1822	446	1174	0	23	3658	2310	778	2159	0	15
	SIRPTS	2518	400	390	970	825	37	3840	400	154	610	741	40	3448	613	148	1424	1357	34
n10m30k3T5	SIRP	6083	8053	807	0	0		8542	8053	4048	0	0		8419	12186	3927	0	0	
	SIRPT	5451	5015	580	0	3367	4	7313	5015	1775	0	3364	15	6252	5245	2660	0	4857	22
	SIRPS	4989	3408	2769	2449	0	9	5867	3408	2032	3442	0	29	7296	6050	1139	4719	0	22
	SIRPTS	6067	1230	54	1813	1955	26	6743	1930	1028	2937	3370	22	6616	1872	626	3307	3886	34
n10m30k3T7	SIRP	8799	12109	198	0	0		10931	12109	3383	0	0		11104	13166	6201	0	0	
	SIRPT	7933	6650	682	0	3503	11	9579	6650	1607	0	4729	15	9564	7773	2396	0	7941	9
	SIRPS	7456	5643	1699	3818	0	12	8765	5643	3095	3859	0	19	9749	7978	2993	6819	0	10
	SIRPTS	7835	1972	402	3201	3124	22	10454	1972	679	2694	2834	29	9245	2858	1537	5063	5556	20
n10m40k3T3	SIRP	3474	4672	1452	0	0		5565	4672	2702	0	0		3971	7715	3693	0	0	
	SIRPT	3365	2565	679	0	1920	11	4504	2565	1378	0	2334	17	5200	4443	887	0	2806	13
	SIRPS	2881	2506	1296	1593	0	14	4369	2506	1497	2191	0	18	3925	3195	637	2865	0	31
	SIRPTS	3766	765	45	1069	1131	29	4804	1065	614	1831	2455	17	5338	1116	218	1452	1628	37
n10m40k3T5	SIRP	8881	9927	2433	0	0		8085	9927	3484	0	0		8370	13368	3432	0	0	
	SIRPT	7636	5959	941	0	3559	15	7827	5959	2086	0	4868	4	8832	7925	1447	0	4987	8
	SIRPS	6751	5054	3273	3449	0	13	8893	5054	3024	3680	0	4	10034	5577	2529	5335	0	7
	SIRPTS	6209	1740	2284	3551	4022	16	8662	1740	1032	2978	4119	14	8085	1890	684	5267	5516	15

n10m40k3T7	SIRP	8773	12146	3765	0	0		11418	12146	3323	0	0		12940	17386	3780	0	0	
	SIRPT	8401	7614	265	0	5240	13	10141	7614	1959	0	5209	7	10757	7854	1998	0	6508	20
	SIRPS	7627	8633	748	3910	0	15	8673	8633	2173	4908	0	9	9820	7786	2693	6907	0	20
	SIRPTS	8291	1549	1108	4086	3676	24	11441	1549	784	4118	4100	18	9247	3505	717	6320	5684	25

Table B.3: Computational results for a number of product varying between 1 and 5 - number of customers equal to 20

Instances	Model	PD					SPD					NBD				
		FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)
n20m1k3T3	SIRP	2148.3	3144.3	0.06	0.04	100	2525.4	3497.6	0.05	0.08	111	2528.8	4313.3	0.03	0.05	137
	SIRPT	1996.8	3119.3	0.02	0.03	128	2246.2	3576.0	0.07	0.02	168	2753.3	2326.5	0.00	0.04	220
n20m1k3T5	SIRP	4536.8	7234.9	0.05	0.07	144	5379.9	8993.3	0.00	0.06	187	6875.9	10831.9	0.07	0.06	236
	SIRPT	4472.6	6487.0	0.05	0.01	124	5575.6	7696.5	0.09	0.02	164	6850.8	9341.1	0.07	0.02	200
n20m1k3T7	SIRP	6348.9	10001.5	0.01	0.07	316	7810.5	10812.6	0.00	0.07	376	9019.3	13682.2	0.09	0.07	500
	SIRPT	5295.2	7601.0	0.02	0.04	217	6068.0	8466.1	0.06	0.09	267	6322.1	11755.4	0.09	0.07	326
n20m3k3T3	SIRP	2652.3	3765.1	0.02	0.03	128	2750.6	4501.1	0.00	0.03	143	3973.9	4996.5	0.07	0.00	189
	SIRPT	2400.8	3598.1	0.10	0.08	147	2939.4	4295.2	0.08	0.07	172	3501.5	4847.3	0.00	0.09	212
	SIRPS	1955.6	2773.8	0.02	0.08	108	2144.3	4110.0	0.05	0.04	132	2304.5	5111.1	0.09	0.01	170
	SIRPTS	2239.7	1577.8	0.03	0.03	102	2588.1	2358.0	0.00	0.09	120	3396.0	2300.5	0.01	0.03	148
n20m3k3T5	SIRP	4959.3	7621.3	0.05	0.06	124	5500.8	8727.8	0.05	0.07	162	6523.8	9668.4	0.00	0.06	198
	SIRPT	4595.6	7185.0	0.01	0.01	104	5317.5	8007.7	0.07	0.05	125	5656.9	9659.5	0.05	0.06	154
	SIRPS	4230.8	6453.1	0.10	0.05	126	5034.0	7113.6	0.03	0.03	155	5755.0	9465.9	0.08	0.05	200
	SIRPTS	4548.8	4546.9	0.03	0.09	135	4660.8	6878.4	0.01	0.03	153	5766.7	8644.0	0.00	0.09	197
n20m3k3T7	SIRP	6981.2	10960.7	0.01	0.10	114	8305.1	12130.7	0.08	0.06	141	10103.6	13091.1	0.05	0.02	174
	SIRPT	6263.4	9286.3	0.07	0.06	331	7270.0	12487.7	0.07	0.02	420	7333.1	13881.8	0.10	0.08	517
	SIRPS	6175.2	9087.3	0.04	0.10	339	7560.4	11121.0	0.06	0.02	400	7406.2	14030.1	0.09	0.09	532
	SIRPTS	6359.1	7213.9	0.03	0.08	217	7372.6	10100.7	0.02	0.02	276	8893.1	9708.5	0.03	0.08	367
n20m5k3T3	SIRP	3105.4	4446.6	0.00	0.07	126	3473.6	5754.9	0.05	0.10	150	4602.8	6499.1	0.06	0.02	190
	SIRPT	2503.4	4019.3	0.00	0.09	250	3065.1	4475.1	0.04	0.03	325	3256.0	6184.4	0.09	0.08	400
	SIRPS	2352.4	3409.0	0.08	0.05	446	2482.6	3901.1	0.08	0.03	535	3408.2	4392.6	0.03	0.04	664
	SIRPTS	2669.0	2160.6	0.03	0.05	403	2977.6	2712.6	0.04	0.03	467	3605.7	2999.3	0.10	0.09	580
n20m5k3T5	SIRP	5997.8	8808.0	0.09	0.02	345	6959.6	10052.3	0.01	0.06	414	8993.2	11455.2	0.01	0.05	505
	SIRPT	5398.3	8261.3	0.09	0.05	209	6166.3	10334.4	0.08	0.01	261	6956.9	12134.5	0.07	0.06	332
	SIRPS	5684.1	5608.3	0.01	0.03	322	5881.3	10183.4	0.08	0.02	406	7367.6	11990.3	0.04	0.08	527
	SIRPTS	5578.5	3821.4	0.02	0.05	326	6435.7	8533.8	0.10	0.05	375	6785.8	11658.9	0.10	0.00	484
n20m5k3T7	SIRP	6319.8	9265.4	0.03	0.01	250	8162.5	11007.3	0.00	0.02	310	7638.9	15211.5	0.07	0.05	378
	SIRPT	6443.6	8236.7	0.01	0.07	341	6705.2	11776.6	0.08	0.03	443	7264.6	14008.0	0.04	0.07	545
	SIRPS	7512.9	7106.5	0.05	0.09	336	8900.1	9108.0	0.08	0.03	407	11253.2	9571.9	0.10	0.04	516
	SIRPTS	6681.8	6504.0	0.06	0.03	148	7428.1	8579.4	0.05	0.01	181	10132.2	9171.0	0.05	0.07	226

Table B.4: Breakdown of cost for a number of product varying between 1 and 5 - number of customers equal to 20

Instances	Model	PD						SPD						NBD					
		T	I	LS	S	Ts	SV (%)	T	I	LS	S	Ts	SV (%)	T	I	LS	S	Ts	SV (%)
n20m1k3T3	SIRP	2148	1903	1241	0	0		2525	2263	1234	0	0		2529	2473	1840	0	0	
	SIRPT	1997	994	991	0	1134	3	2246	1665	589	0	1322	3	2753	1465	98	0	763	26
n20m1k3T5	SIRP	4537	4961	2274	0	0		5380	5143	3850	0	0		6876	6399	4432	0	0	
	SIRPT	4473	2574	1836	0	2077	7	5576	3284	2025	0	2388	8	6851	5821	8	0	3512	9
n20m1k3T7	SIRP	6349	5995	4007	0	0		7811	7174	3638	0	0		9019	7189	6494	0	0	
	SIRPT	5295	2688	2043	0	2870	21	6068	3233	2456	0	2778	22	6322	7602	256	0	3897	20
n20m3k3T3	SIRP	2652	1888	1877	0	0		2751	3370	1132	0	0		3974	3831	1165	0	0	
	SIRPT	2401	1102	1058	0	1439	7	2939	1729	1012	0	1554	0	3501	2702	486	0	1659	7
	SIRPS	1956	1028	848	898	0	26	2144	1539	1165	1406	0	14	2304	3069	450	1592	0	17
	SIRPTS	2240	239	180	560	599	41	2588	327	177	889	966	32	3396	528	121	758	893	36
n20m3k3T5	SIRP	4959	3944	3677	0	0		5501	5488	3240	0	0		6524	5901	3768	0	0	
	SIRPT	4596	2846	1948	0	2391	6	5318	3796	1404	0	2808	6	5657	5509	434	0	3716	5
	SIRPS	4231	2345	1762	2347	0	15	5034	2989	1795	2330	0	15	5755	5717	712	3037	0	6
	SIRPTS	4549	1020	454	1334	1740	28	4661	1108	695	2594	2482	19	5767	2715	196	2714	3019	11
n20m3k3T7	SIRP	6981	5487	5474	0	0		8305	8411	3719	0	0		10104	8480	4611	0	0	
	SIRPT	6263	3527	2151	0	3609	13	7270	4086	3302	0	5100	3	7333	8479	312	0	5090	9
	SIRPS	6175	3982	2299	2806	0	15	7560	4484	2909	3728	0	9	7406	8048	1880	4102	0	8
	SIRPTS	6359	1068	953	2467	2725	24	7373	2348	1139	3142	3471	14	8893	2901	92	3454	3261	20
n20m5k3T3	SIRP	3105	2859	1588	0	0		3474	3835	1920	0	0		4603	3383	3116	0	0	
	SIRPT	2503	1516	1206	0	1298	14	3065	1618	1216	0	1641	18	3256	4063	35	0	2087	15
	SIRPS	2352	1372	804	1233	0	24	2483	1584	886	1432	0	31	3408	2879	8	1505	0	30
	SIRPTS	2669	388	264	649	859	36	2978	436	328	1005	944	38	3606	801	76	989	1133	41
n20m5k3T5	SIRP	5998	5985	2823	0	0		6960	5944	4108	0	0		8993	7849	3606	0	0	
	SIRPT	5398	3066	2320	0	2875	8	6166	3769	2906	0	3659	3	6957	7236	726	0	4173	7
	SIRPS	5684	1891	1835	1882	0	24	5881	4003	2813	3368	0	6	7368	7215	701	4074	0	5
	SIRPTS	5579	791	377	1301	1352	37	6436	1357	664	3023	3490	12	6786	2755	798	3992	4114	10
n20m5k3T7	SIRP	6320	5617	3649	0	0		8162	8145	2862	0	0		7639	11092	4119	0	0	
	SIRPT	6444	3466	1754	0	3017	6	6705	4566	3100	0	4111	4	7265	8392	564	0	5051	7
	SIRPS	7513	2808	1703	2595	0	6	8900	4226	1482	3401	0	6	11253	5071	995	3505	0	9
	SIRPTS	6682	1013	987	2264	2240	15	7428	1663	1236	2651	3029	16	10132	2932	92	2741	3406	16

Table B.5: Computational results for a number of product varying between 20 and 40 - number of customers equal to 20

Instances	Model	PD					SPD					NBD				
		FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)
n20m20k3T3	SIRP	2958.4	5631.5	0.43	0.40	1043	3941.8	6339.4	0.53	0.41	1026	4141.4	7880.2	0.51	0.81	915
	SIRPT	2847.9	4215.2	0.40	0.42	997	4378.4	4839.6	0.27	0.41	1160	4385.2	6461.1	0.56	0.58	989
	SIRPS	2903.9	4111.3	0.31	0.45	1040	3603.2	4955.5	0.62	0.43	913	3234.0	6601.8	0.35	0.67	1193
	SIRPTS	2474.4	3327.2	0.68	0.34	893	2764.5	4440.3	0.40	0.23	1154	3706.0	5141.0	0.49	0.66	1142
n20m20k3T5	SIRP	6431.9	11908.3	0.70	0.71	1064	8452.6	13265.3	0.22	0.76	1049	12678.7	17566.3	0.98	0.83	1062
	SIRPT	6810.6	9601.5	0.75	0.52	1037	9215.2	11263.1	0.65	0.23	1064	11569.0	15167.8	0.36	0.45	1050
	SIRPS	6429.0	9611.4	0.55	0.45	984	6422.1	13581.1	0.54	0.85	1022	6994.6	19494.3	0.78	0.85	903
	SIRPTS	4156.7	7126.4	0.74	0.52	832	4706.3	13948.7	0.46	0.30	952	5961.3	9926.9	0.51	0.90	1162
n20m20k3T7	SIRP	9519.6	17213.3	0.69	0.50	880	12545.3	17568.2	0.21	0.50	831	13339.4	21507.5	0.89	0.36	1138
	SIRPT	8323.2	16419.8	0.41	0.77	1147	9833.0	16418.6	0.66	0.84	991	12108.8	17657.2	0.51	0.84	1117
	SIRPS	7561.6	16707.5	0.66	0.31	814	10024.3	16485.0	0.89	0.40	886	9176.0	15847.3	0.70	0.31	1127
	SIRPTS	9367.0	11725.0	0.49	0.42	822	12722.8	7919.8	0.86	0.42	981	14022.7	4459.0	0.51	0.44	1054
n20m30k3T3	SIRP	3865.2	6922.4	0.70	0.50	1020	5119.3	10657.4	0.75	0.39	916	5673.0	9271.6	0.45	0.99	1190
	SIRPT	3591.7	5168.4	0.72	0.51	1173	5201.9	8232.8	0.68	0.88	912	4817.0	8449.3	0.50	0.69	1066
	SIRPS	2893.2	5127.8	0.36	0.32	1045	3626.6	8380.9	0.24	0.64	1074	3533.0	6821.6	0.31	0.43	1166
	SIRPTS	3765.5	1862.5	0.41	0.62	1136	5915.4	4997.5	0.38	0.42	854	5255.2	2946.4	0.49	0.87	991
n20m30k3T5	SIRP	8553.3	13802.4	0.56	0.42	955	8965.1	16477.4	0.79	0.74	852	10691.1	18017.6	0.56	0.42	1188
	SIRPT	7106.7	11930.8	0.48	0.44	814	10080.7	13902.5	0.38	0.45	913	9861.3	15739.2	0.64	0.94	1163
	SIRPS	7058.2	12448.4	0.75	0.32	857	7966.4	15502.3	0.68	0.70	1197	10895.0	14619.9	0.63	0.48	1130
	SIRPTS	8094.3	8072.9	0.40	0.64	1068	8906.2	12059.4	0.76	0.73	1019	9515.6	12191.2	0.65	0.50	868
n20m30k3T7	SIRP	11753.1	20219.4	0.75	0.36	1174	14677.0	21269.6	0.67	0.89	845	12368.5	23142.5	0.39	0.78	1143
	SIRPT	10508.4	14153.4	0.39	0.65	916	12985.2	18517.0	0.83	0.26	1051	11922.3	21399.1	0.79	0.64	1171
	SIRPS	9329.5	14052.7	0.39	0.32	993	12807.9	16615.3	0.53	0.74	929	16166.1	17379.6	0.88	0.50	830
	SIRPTS	10583.6	10877.8	0.34	0.66	856	12787.6	16006.5	0.34	0.36	1058	14361.1	14776.5	0.79	0.90	808
n20m40k3T3	SIRP	4880.4	6817.6	0.40	0.61	904	5762.6	8320.1	0.88	0.34	1109	7379.8	11326.9	0.91	0.38	1155
	SIRPT	3431.4	6646.2	0.55	0.53	1187	4722.1	6588.3	0.56	0.67	868	5462.6	10907.1	0.82	0.39	1194
	SIRPS	3647.1	6464.2	0.31	0.60	875	4339.8	7516.6	0.41	0.71	1193	4749.6	11274.8	0.71	0.67	1099
	SIRPTS	3808.1	4723.2	0.62	0.36	938	5666.3	3084.1	0.70	0.66	1195	5832.3	8265.7	0.99	0.43	944
n20m40k3T5	SIRP	8509.2	14958.0	0.45	0.38	1200	12493.7	18195.8	0.87	0.60	975	11228.6	23267.8	0.86	0.86	1051
	SIRPT	8576.5	13276.0	0.53	0.39	1197	11335.0	14257.7	0.41	0.59	1080	12095.7	20161.6	0.98	0.62	849
	SIRPS	8638.7	12430.2	0.40	0.37	1020	10451.9	13837.9	0.78	0.69	908	11663.8	20689.9	0.49	0.33	953
	SIRPTS	8629.7	10531.8	0.60	0.41	1137	11789.2	9291.2	0.58	0.82	1171	11253.4	17526.9	0.94	0.99	1028

n20m40k3T7	SIRP	9473.6	18049.9	0.57	0.41	966	14880.7	19701.5	0.86	0.56	985	15051.5	25279.4	0.34	0.60	1142
	SIRPT	10390.7	14999.2	0.44	0.60	916	13507.1	18614.4	0.25	0.28	816	12992.1	20278.4	0.67	0.71	935
	SIRPS	12174.2	13635.4	0.78	0.44	963	17114.2	15116.6	0.25	0.40	1186	13608.4	19856.6	0.65	0.55	1049
	SIRPTS	10488.8	11805.5	0.63	0.78	1077	12096.0	15890.4	0.28	0.55	1186	13406.7	16509.7	0.92	0.96	893

Table B.6: Breakdown of cost for a number of product varying between 20 and 40 - number of customers equal to 20

Instances	Model	PD						SPD						NBD					
		T	I	LS	S	Ts	SV (%)	T	I	LS	S	Ts	SV (%)	T	I	LS	S	Ts	SV (%)
n20m20k3T3	SIRP	2958	4971	661	0	0		3942	4971	1369	0	0		4141	5175	2705	0	0	
	SIRPT	2848	1945	959	0	1311	18	4378	1945	1106	0	1789	10	4385	2838	1176	0	2447	10
	SIRPS	2904	1990	526	1596	0	18	3603	1990	1133	1833	0	17	3234	3557	724	2321	0	18
	SIRPTS	2474	960	139	1128	1100	32	2764	960	534	1375	1571	30	3706	1162	433	1704	1843	26
n20m20k3T5	SIRP	6432	10186	1722	0	0		8453	10186	3079	0	0		12679	12866	4701	0	0	
	SIRPT	6811	5139	1119	0	3343	11	9215	5139	1987	0	4137	6	11569	7243	2251	0	5674	12
	SIRPS	6429	6046	868	2697	0	13	6422	6046	2667	4867	0	8	6995	10121	2369	7004	0	12
	SIRPTS	4157	1732	272	2640	2482	38	4706	2832	962	5387	4768	14	5961	1723	618	3870	3717	47
n20m20k3T7	SIRP	9520	13723	3491	0	0		12545	13723	3846	0	0		13339	14468	7039	0	0	
	SIRPT	8323	7845	3479	0	5095	7	9833	7845	2927	0	5646	13	12109	8385	2656	0	6616	15
	SIRPS	7562	6949	4689	5069	0	9	10024	6949	3003	6533	0	12	9176	6953	3242	5653	0	28
	SIRPTS	9367	1781	1023	4351	4570	21	12723	1781	566	2455	3118	31	14023	663	295	1720	1781	47
n20m30k3T3	SIRP	3865	6552	370	0	0		5119	8252	2405	0	0		5673	6498	2774	0	0	
	SIRPT	3592	3216	176	0	1776	19	5202	3816	1409	0	3007	15	4817	3368	1809	0	3273	11
	SIRPS	2893	3051	47	2030	0	26	3627	3451	1669	3260	0	24	3533	3453	901	2468	0	31
	SIRPTS	3766	584	45	639	595	48	5915	1284	490	1608	1617	31	5255	345	141	1193	1267	45
n20m30k3T5	SIRP	8553	10901	2901	0	0		8965	10901	5576	0	0		10691	14084	3933	0	0	
	SIRPT	7107	6040	1693	0	4198	15	10081	6040	2182	0	5681	6	9861	8142	1921	0	5676	11
	SIRPS	7058	7779	182	4488	0	13	7966	7979	2718	4806	0	8	10895	7550	2197	4873	0	11
	SIRPTS	8094	2560	29	2298	3186	28	8906	2660	1087	4195	4117	18	9516	2537	621	4629	4403	24
n20m30k3T7	SIRP	11753	13379	6841	0	0		14677	13379	7891	0	0		12368	15281	7862	0	0	
	SIRPT	10508	8487	31	0	5636	23	12985	8487	2581	0	7449	12	11922	9829	3085	0	8485	6
	SIRPS	9330	8897	352	4803	0	27	12808	8897	2237	5481	0	18	16166	8491	2164	6724	0	6
	SIRPTS	10584	3271	44	3620	3943	33	12788	3671	853	4896	6587	20	14361	1805	320	6181	6471	18
n20m40k3T3	SIRP	4880	6174	643	0	0		5763	6174	2146	0	0		7380	7370	3956	0	0	
	SIRPT	3431	3112	1184	0	2351	14	4722	3112	1029	0	2448	20	5463	4677	2385	0	3845	12
	SIRPS	3647	3186	701	2577	0	14	4340	3186	1427	2903	0	16	4750	5409	1591	4274	0	14
	SIRPTS	3808	789	437	1847	1650	27	5666	789	191	1043	1062	38	5832	868	285	3501	3612	25
n20m40k3T5	SIRP	8509	12515	2443	0	0		12494	12515	5681	0	0		11229	19296	3972	0	0	
	SIRPT	8577	5211	3284	0	4781	7	11335	5211	3070	0	5977	17	12096	9463	3164	0	7534	6
	SIRPS	8639	6429	1989	4012	0	10	10452	6429	2590	4818	0	21	11664	8078	4167	8446	0	6
	SIRPTS	8630	1453	1879	4017	3183	18	11789	1453	493	3645	3700	31	11253	3938	1145	6376	6068	17

n20m40k3T7	SIRP	9474	12376	5673	0	0		14881	12376	7325	0	0		15051	19263	6016	0	0	
	SIRPT	10391	7464	2427	0	5109	8	13507	7464	3847	0	7304	7	12992	9915	2639	0	7724	18
	SIRPS	12174	6872	2448	4316	0	6	17114	6872	3443	4801	0	7	13608	8403	4426	7027	0	17
	SIRPTS	10489	2875	840	4087	4003	19	12096	2875	1524	5371	6121	19	13407	2521	1311	6993	5684	26

Table B.7: Computational results for a number of product varying between 1 and 5 - number of customers equal to 50

Instances	Model	PD					SPD					NBD				
		FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)
n50m1k3T3	SIRP	2714.2	4007.4	0.04	0.03	123	2942.9	4894.5	0.07	0.08	137	3562.7	5850.0	0.05	0.07	183
	SIRPT	2879.7	3341.1	0.08	0.08	814	3345.0	3123.4	0.04	0.03	952	4495.1	3523.0	0.10	0.00	1138
n50m1k3T5	SIRP	5987.4	8727.2	0.04	0.08	837	7364.0	10676.1	0.02	0.00	937	7139.9	13822.7	0.08	0.09	1172
	SIRPT	4566.5	6612.3	0.07	0.01	910	5247.4	9435.5	0.08	0.06	1147	5371.1	13063.1	0.04	0.08	1114
n50m1k3T7	SIRP	7779.3	12495.2	0.05	0.03	1022	9267.2	14271.5	0.02	0.09	1018	9898.4	19266.0	0.05	0.05	1153
	SIRPT	5650.9	8406.0	0.04	0.02	1037	6167.3	12209.0	0.01	0.04	1196	6555.2	12179.3	0.05	0.05	1198
n50m3k3T3	SIRP	3221.0	4608.3	0.07	0.06	1020	3315.0	5485.1	0.02	0.10	1153	4724.5	8950.0	0.05	0.05	1133
	SIRPT	2987.7	4690.8	0.06	0.09	1044	3518.1	3749.9	0.02	0.08	1105	4984.4	6267.0	0.06	0.01	1131
	SIRPS	2252.0	3234.0	0.02	0.08	1031	2402.3	4235.8	0.01	0.03	1155	2923.3	5334.6	0.02	0.06	1136
	SIRPTS	2784.4	1338.5	0.01	0.02	733	3490.3	1796.3	0.07	0.01	814	3708.1	3672.4	0.03	0.05	1066
n50m3k3T5	SIRP	7491.9	10749.9	0.01	0.01	944	8989.5	13411.5	0.09	0.07	1046	9362.7	16107.2	0.01	0.07	1157
	SIRPT	6773.8	10190.3	0.04	0.00	1018	7451.0	11996.2	0.00	0.06	1022	9074.0	14842.8	0.05	0.01	1051
	SIRPS	5775.1	8327.7	0.07	0.05	1021	6314.3	10862.8	0.09	0.09	1044	8842.2	10877.2	0.01	0.02	1175
	SIRPTS	5666.7	8104.1	0.07	0.03	900	5867.6	10368.2	0.07	0.05	990	7959.9	10351.1	0.04	0.04	1138
n50m3k3T7	SIRP	9468.7	14394.0	0.02	0.04	931	10872.8	17762.5	0.01	0.05	1192	12500.6	11775.8	0.07	0.01	1154
	SIRPT	6830.4	9652.2	0.01	0.08	1020	7477.7	19999.4	0.06	0.00	1065	8971.0	13885.1	0.05	0.06	1156
	SIRPS	8163.8	8593.3	0.05	0.01	1107	9693.2	18173.9	0.05	0.02	1051	12214.8	8257.9	0.02	0.04	1114
	SIRPTS	6977.8	8589.6	0.08	0.08	1100	8249.6	13200.2	0.01	0.02	1009	11887.5	6989.1	0.00	0.02	1141
n50m5k3T3	SIRP	4322.9	6098.8	0.06	0.03	1103	5296.7	7365.7	0.04	0.08	1157	6257.0	8355.4	0.03	0.09	1047
	SIRPT	3165.3	4661.9	0.02	0.09	1149	3642.2	6774.0	0.09	0.09	1167	4529.3	6120.4	0.04	0.09	1136
	SIRPS	2792.4	4121.3	0.06	0.02	1031	2942.8	6772.9	0.07	0.08	1189	3887.5	5610.5	0.07	0.01	1088
	SIRPTS	3543.3	1676.8	0.04	0.08	1111	4330.4	2977.6	0.04	0.01	1189	5735.5	1484.6	0.01	0.03	1027
n50m5k3T5	SIRP	8734.7	13474.1	0.04	0.08	1035	10493.1	16357.3	0.01	0.00	1142	11211.8	20767.1	0.02	0.05	1002
	SIRPT	6379.8	13192.1	0.07	0.00	1032	7388.0	16391.8	0.01	0.05	1142	10282.9	19097.9	0.03	0.02	1031
	SIRPS	8089.8	10919.6	0.08	0.01	1138	10541.2	10990.3	0.00	0.10	1134	9170.9	18917.7	0.03	0.01	1164
	SIRPTS	6270.5	9713.4	0.02	0.01	1037	8190.7	11149.8	0.06	0.03	1038	8765.5	13727.5	0.01	0.09	1093
n50m5k3T7	SIRP	6842.1	16925.0	0.00	0.09	1110	7953.2	18420.6	0.04	0.03	1143	8572.8	16798.0	0.06	0.04	1079
	SIRPT	9156.9	11862.0	0.01	0.03	1063	10719.7	14406.8	0.08	0.06	1144	12886.1	9144.9	0.00	0.08	1067
	SIRPS	9966.5	10169.7	0.07	0.08	1012	10551.7	15399.7	0.06	0.10	1126	13102.6	8271.4	0.08	0.10	984
	SIRPTS	8516.5	11309.5	0.01	0.03	1025	9476.7	13754.1	0.05	0.07	1171	10727.6	8792.5	0.00	0.06	1127

Table B.8: Breakdown of cost for a number of product varying between 1 and 5 - number of customers equal to 50

Instances	Model	PD						SPD						NBD					
		T	I	LS	S	Ts	SV (%)	T	I	LS	S	Ts	SV (%)	T	I	LS	S	Ts	SV (%)
n50m1k3T3	SIRP	2714	2513	1495	0	0		2943	3524	1371	0	0		3563	3799	2051	0	0	
	SIRPT	2880	1256	809	0	1276	7	3345	1332	688	0	1103	17	4495	2352	20	0	1151	15
n50m1k3T5	SIRP	5987	6024	2704	0	0		7364	6859	3817	0	0		7140	8158	5665	0	0	
	SIRPT	4567	2778	1252	0	2582	24	5247	4437	1735	0	3264	19	5371	4253	305	3671	4835	12
n50m1k3T7	SIRP	7779	8067	4428	0	0		9267	9062	5209	0	0		9898	12912	6354	0	0	
	SIRPT	5651	2785	2295	0	3326	31	6167	5335	2970	0	3904	22	6555	7353	122	0	4705	36
n50m3k3T3	SIRP	3221	2548	2060	0	0		3315	3208	2277	0	0		4725	6655	2295	0	0	
	SIRPT	2988	2186	975	0	1530	2	3518	1306	915	0	1529	17	4984	4075	65	0	2127	18
	SIRPS	2252	1236	971	1027	0	30	2402	1986	675	1575	0	25	2923	3324	443	1567	0	40
	SIRPTS	2784	269	158	466	446	47	3490	228	181	659	728	40	3708	1083	107	1069	1413	46
n50m3k3T5	SIRP	7492	6435	4315	0	0		8990	7413	5999	0	0		9363	10507	5600	0	0	
	SIRPT	6774	4141	2748	0	3301	7	7451	5054	2147	0	4795	13	9074	8581	575	0	5686	6
	SIRPS	5775	3725	2191	2412	0	23	6314	4811	2489	3563	0	23	8842	7309	353	3215	0	23
	SIRPTS	5667	1780	841	2382	3101	25	5868	2137	1461	3240	3530	28	7960	2804	208	3653	3686	28
n50m3k3T7	SIRP	9469	8841	5553	0	0		10873	12443	5320	0	0		12501	6682	5094	0	0	
	SIRPT	6830	3729	2659	0	3264	31	7478	8653	4191	0	7156	4	8971	7435	1998	0	4452	6
	SIRPS	8164	4035	2084	2474	0	30	9693	7550	4954	5670	0	3	12215	5296	87	2875	0	16
	SIRPTS	6978	1467	1116	2573	3434	35	8250	1965	1362	4656	5217	25	11888	1997	303	2494	2195	22
n50m5k3T3	SIRP	4323	3590	2509	0	0		5297	5268	2098	0	0		6257	5102	3254	0	0	
	SIRPT	3165	1982	926	0	1754	25	3642	2647	1716	0	2411	18	4529	4107	14	0	1999	27
	SIRPS	2792	1402	1305	1414	0	34	2943	2919	1686	2169	0	23	3888	2868	1179	1564	0	35
	SIRPTS	3543	338	213	491	634	50	4330	637	246	984	1111	42	5735	521	0	478	485	51
n50m5k3T5	SIRP	8735	7944	5530	0	0		10493	9567	6790	0	0		11212	13719	7048	0	0	
	SIRPT	6380	5066	3415	0	4711	12	7388	7331	4083	0	4978	11	10283	10756	2085	0	6256	8
	SIRPS	8090	4452	2532	3935	0	14	10541	4199	3002	3788	0	20	9171	12209	452	6256	0	12
	SIRPTS	6270	1856	1241	3176	3440	28	8191	2735	926	3717	3771	28	8766	5194	313	3816	4404	30
n50m5k3T7	SIRP	6842	10509	6416	0	0		7953	12213	6208	0	0		8573	9850	6948	0	0	
	SIRPT	9157	3996	3592	0	4274	12	10720	5453	3467	0	5486	5	12886	5779	5	0	3361	13
	SIRPS	9966	3657	3614	2898	0	15	10552	7517	3343	4540	0	2	13103	4725	674	2873	0	16
	SIRPTS	8517	2107	1279	3927	3997	17	9477	2742	2203	4382	4427	12	10728	2231	88	3178	3296	23

Table B.g: Computational results for a number of product varying between 20 and 40 - number of customers equal to 50

Instances	Model	PD					SPD					NBD				
		FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)	FSC	SSC	LB (%)	UB (%)	CPU (s)
n50m20k3T3	SIRP	4189.7	7894.2	0.60	0.34	1140	6486.8	7651.9	0.62	0.68	884	5681.2	10979.3	0.82	0.94	929
	SIRPT	5310.2	5234.5	0.70	0.32	1136	5750.2	5095.7	0.69	0.29	827	7075.3	7787.1	0.60	0.86	1154
	SIRPS	3097.9	7374.1	0.33	0.66	1043	4360.7	6286.9	0.70	0.24	811	3956.6	10154.3	0.94	0.47	1178
	SIRPTS	3245.0	5371.6	0.75	0.66	1043	4627.2	3995.4	0.23	0.24	1070	4455.3	5757.7	0.75	0.50	1091
n50m20k3T5	SIRP	8704.8	16059.9	0.70	0.72	1142	12703.4	15818.0	0.49	0.73	1055	10128.5	21336.3	0.75	0.32	1175
	SIRPT	7578.3	10173.6	0.71	0.63	931	9572.6	14180.8	0.30	0.32	866	10322.3	16579.6	0.97	0.65	924
	SIRPS	7760.2	11846.2	0.70	0.74	1152	7831.4	11172.2	0.71	0.66	1118	8818.3	15315.0	0.71	0.56	986
	SIRPTS	5039.3	7919.1	0.66	0.58	948	6299.6	7467.0	0.63	0.24	934	6197.2	10926.8	0.67	0.98	913
n50m20k3T7	SIRP	12111.8	20854.5	0.59	0.65	1030	19547.9	23480.8	0.65	0.62	983	19948.0	32285.5	0.60	0.74	1150
	SIRPT	8820.1	15357.8	0.78	0.36	810	13174.9	21122.2	0.66	0.35	898	13650.1	28966.6	0.68	0.76	1049
	SIRPS	8544.7	15066.0	0.64	0.47	852	12009.2	23327.8	0.33	0.58	1195	13411.9	31137.1	0.33	0.43	1096
	SIRPTS	12904.1	7019.0	0.71	0.50	1021	15048.4	13507.6	0.56	0.39	1026	16133.6	22243.0	0.92	0.99	1107
n50m30k3T3	SIRP	5320.9	7918.4	0.55	0.50	803	5063.2	8392.1	0.60	0.24	926	6508.0	12247.1	0.79	0.78	953
	SIRPT	4664.1	6921.1	0.77	0.34	1084	5803.2	5516.8	0.71	0.28	1164	6491.8	10613.0	0.34	0.56	827
	SIRPS	3385.1	7601.1	0.63	0.72	988	5113.1	6151.8	0.64	0.62	1054	4943.0	12376.9	0.81	0.44	1112
	SIRPTS	4268.5	4524.2	0.33	0.75	834	5505.4	4363.7	0.47	0.89	970	5809.2	10019.4	0.80	0.56	1166
n50m30k3T5	SIRP	11174.6	17775.2	0.45	0.56	824	13835.3	20124.6	0.28	0.55	930	18103.9	25551.5	0.53	0.85	874
	SIRPT	10564.5	15562.8	0.79	0.55	904	14373.3	15844.5	0.42	0.39	1127	14554.8	24931.9	0.62	0.34	1008
	SIRPS	9375.0	15826.6	0.75	0.43	920	12109.5	18837.7	0.76	0.58	806	12168.8	27337.1	0.86	0.95	930
	SIRPTS	8392.5	14147.0	0.76	0.70	831	11184.4	15221.6	0.65	0.83	875	11221.4	21276.1	0.44	0.82	935
n50m30k3T7	SIRP	13231.7	24757.7	0.33	0.46	1069	19379.8	28412.5	0.42	0.74	1072	17471.9	36856.2	0.74	0.87	827
	SIRPT	9669.3	22944.7	0.51	0.32	1097	14361.6	29783.9	0.59	0.40	1061	12877.2	36526.7	0.58	0.97	895
	SIRPS	13086.7	19611.9	0.49	0.77	851	14319.7	29980.9	0.81	0.33	1062	20766.6	30057.2	0.65	0.63	954
	SIRPTS	11635.0	15735.0	0.71	0.61	1180	16101.0	21221.6	0.58	0.55	1058	16069.3	29056.5	0.47	0.68	1107
n50m40k3T3	SIRP	6394.7	30832.4	0.79	0.33	1052	9187.8	11325.3	0.27	0.65	1029	8283.4	14219.7	0.60	0.39	1123
	SIRPT	4597.6	29865.3	0.68	0.31	1095	6156.1	9211.1	0.47	0.78	938	7267.8	10602.4	0.54	0.90	1136
	SIRPS	4353.9	30514.5	0.46	0.77	1090	4723.4	10529.2	0.54	0.74	897	5821.6	11018.4	0.91	0.92	1173
	SIRPTS	5749.8	22620.0	0.57	0.38	829	6654.8	5941.2	0.52	0.41	1178	7115.2	7282.9	0.80	0.78	1030
n50m40k3T5	SIRP	15555.0	21666.9	0.79	0.36	1178	19721.5	28662.9	0.63	0.38	1012	18428.7	35519.6	0.43	0.60	892
	SIRPT	9705.8	25022.4	0.77	0.44	805	12491.4	27053.8	0.29	0.61	931	15974.7	26064.0	0.62	0.35	806
	SIRPS	13219.6	20256.1	0.49	0.39	951	14697.8	24748.8	0.90	0.83	923	15173.7	28807.3	0.92	0.35	881
	SIRPTS	8657.2	15958.0	0.51	0.66	1085	15280.9	19615.0	0.56	0.86	923	13711.9	21692.0	0.44	0.95	912

n50m40k3T7	SIRP	11093.0	38880.1	0.79	0.45	896	15387.4	29067.2	0.86	0.86	985	14070.6	27968.7	0.99	0.55	805
	SIRPT	14619.5	21612.2	0.78	0.66	1059	16179.7	25074.0	0.52	0.30	1089	19213.9	18481.0	0.79	0.70	810
	SIRPS	16061.8	20402.0	0.52	0.52	1015	19098.4	22608.4	0.33	0.25	856	23291.7	14112.8	0.72	0.65	1123
	SIRPTS	12113.8	20498.2	0.73	0.42	956	17007.5	20894.2	0.67	0.59	965	18137.7	12541.0	0.46	0.63	864

Table B.10: Breakdown of cost for a number of product varying between 20 and 40 - number of customers equal to 50

Instances	Model	PD						SPD						NBD					
		T	I	LS	S	Ts	SV (%)	T	I	LS	S	Ts	SV (%)	T	I	LS	S	Ts	SV (%)
n50m20k3T3	SIRP	4190	5274	2620	0	0		6487	5274	2377	0	0		5681	8769	2210	0	0	
	SIRPT	5310	2046	1300	0	1889	13	5750	2046	1122	0	1927	23	7075	4078	1032	0	2677	11
	SIRPS	3098	2550	2286	2538	0	13	4361	2550	1321	2416	0	25	3957	3987	2002	4165	0	15
	SIRPTS	3245	828	714	1815	2015	29	4627	828	310	1258	1600	39	4455	955	425	2387	1991	39
n50m20k3T5	SIRP	8705	10013	6047	0	0		12703	10013	5805	0	0		10129	15134	6202	0	0	
	SIRPT	7578	6588	183	0	3403	28	9573	6588	1894	0	5699	17	10322	7895	2653	0	6032	15
	SIRPS	7760	5896	2218	3733	0	21	7831	5896	1666	3611	0	33	8818	8128	2162	5025	0	23
	SIRPTS	5039	1591	493	2928	2907	48	6300	1591	403	2784	2689	52	6197	1423	477	4483	4543	46
n50m20k3T7	SIRP	12112	15706	5148	0	0		19548	15706	7774	0	0		19948	23191	9095	0	0	
	SIRPT	8820	8345	1392	0	5621	27	13175	8345	4107	0	8671	20	13650	13005	4073	0	11888	18
	SIRPS	8545	10404	18	4643	0	28	12009	11404	4841	7082	0	18	13412	14582	5657	10898	0	15
	SIRPTS	12904	1833	473	2424	2289	40	15048	1833	997	5209	5469	34	16134	2990	1525	8490	9238	27
n50m30k3T3	SIRP	5321	6493	1425	0	0		5063	6493	1899	0	0		6508	10406	1841	0	0	
	SIRPT	4664	2261	2237	0	2423	12	5803	2261	1296	0	1960	16	6492	4989	1985	0	3639	9
	SIRPS	3385	2813	2207	2581	0	17	5113	2813	1279	2060	0	16	4943	5807	1402	5167	0	8
	SIRPTS	4269	623	639	1554	1707	34	5505	623	280	1723	1737	27	5809	1607	560	4253	3600	16
n50m30k3T5	SIRP	11175	15411	2364	0	0		13835	15411	4713	0	0		18104	20257	5294	0	0	
	SIRPT	10564	6851	2661	0	6051	10	14373	6851	2586	0	6407	11	14555	11552	3342	0	10038	10
	SIRPS	9375	9491	131	6204	0	13	12110	9891	3274	5672	0	9	12169	12838	4726	9773	0	10
	SIRPTS	8393	2831	2394	4312	4611	22	11184	2831	1191	5978	5223	22	11221	3365	1631	6976	9304	26
n50m30k3T7	SIRP	13232	19213	5545	0	0		19380	19213	9200	0	0		17472	26817	10040	0	0	
	SIRPT	9669	15055	926	0	6964	14	14362	15055	3554	0	11175	8	12877	19994	4040	0	12492	9
	SIRPS	13087	12128	63	7421	0	14	14320	13628	6933	9420	0	7	20767	12952	5158	11948	0	6
	SIRPTS	11635	4099	3	5926	5707	28	16101	4505	1307	7058	8351	22	16069	5168	2116	10452	11320	17
n50m40k3T3	SIRP	6395	7801	23032	0	0		9188	7801	3524	0	0		8283	9288	4931	0	0	
	SIRPT	4598	3837	14578	0	11450	7	6156	3837	1768	0	3607	25	7268	5260	994	0	4349	21
	SIRPS	4354	4575	13788	12151	0	6	4723	4575	2177	3777	0	26	5822	5353	1703	3962	0	25
	SIRPTS	5750	1311	3606	8849	8853	24	6655	1311	529	2183	1918	39	7115	1493	389	2499	2902	36
n50m40k3T5	SIRP	15555	18339	3328	0	0		19721	18339	10324	0	0		18429	23539	11981	0	0	
	SIRPT	9706	11191	6087	0	7744	7	12491	11191	5393	0	10470	18	15975	10019	5002	0	11043	22
	SIRPS	13220	9977	2249	8030	0	10	14698	9977	6102	8670	0	18	15174	12815	4740	11252	0	18
	SIRPTS	8657	2973	1353	5616	6016	34	15281	2973	1230	7650	7762	28	13712	3173	1399	9228	7892	34

n50m40k3T7	SIRP	11093	22341	16539	0	0		15387	22341	6726	0	0		14071	19103	8866	0	0	
	SIRPT	14620	9245	4464	0	7904	27	16180	9245	5436	0	10393	7	19214	7052	3508	0	7921	10
	SIRPS	16062	10078	3783	6541	0	27	19098	10078	3508	9023	0	6	23292	7161	1719	5233	0	11
	SIRPTS	12114	2640	3469	6533	7857	35	17008	2640	1222	8314	8719	15	18138	2220	615	4461	5245	27