Optimization strategies for the integrated routing and inventory management problem

Diego Perdigão Martino

UNIVERSITÉ CLERMONT AUVERGNE

ÉCOLE DOCTORALE DES SCIENCES POUR L'INGÉNIEUR

LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION
ET D'OPTIMISATION DES SYSTÈMES

Thesis presented by | *Thèse présentée par*

Diego Perdigão Martino

to obtain the degree of | *pour obtenir le grade de*

Docteur en Informatique

Optimization strategies for the integrated routing and inventory management problem

Stratégies d'optimisation pour le problème intégré de transport et de gestion de stock

Publicly defended on November 27^{th}, 2024, to the jury
Soutenue publiquement le 27 novembre 2024 devant le jury

Caroline Prodhon	Présidente du jury, Rapporteure, Professeure des Universités, Université de Technologie de Troyes, France
Marie-Ange Manier	Rapporteure, Professeure des Universités, Université de Technologie de Belfort Montbéliard, France
Alexandre Xavier Martins	Examinateur, Professor Associado, Universidade Federal de Ouro Preto, Brésil
Laurent Deroussi	Invité, Maître de Conférences HDR, Université Clermont Auvergne, France
Philippe Lacomme	Directeur, Professeur des Universités, Université Clermont Auvergne, France
Katyanne Farias	Encadrante, Maîtresse de Conférences, Université Clermont Auvergne, France
Manuel Iori	Encadrant, Professore Ordinario, Università degli Studi di Modena e Reggio Emilia, Italie

(EN) Thesis carried out at the Doctoral School *Sciences pour l'Ingénieur* (ED SPI) and the *Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes* (LIMOS), *Unité Mixte de Recherche* (UMR 6158), Cézeaux University Campus, 1 rue de la Chebarde, 63178, Aubière, France.

(FR) Thèse préparée au sein de l'École Doctorale Sciences pour l'Ingénieur (ED SPI) et du Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS), Unité Mixte de Recherche (UMR 6158), Campus Universitaire des Cézeaux, 1 rue de la Chebarde, 63178, Aubière, France.

(PT-BR) Tese realizada na Escola de Pos-Graduação *Sciences pour l'Ingénieur* (ED SPI) e no *Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes* (LIMOS), *Unité Mixte de Recherche* (UMR 6158), Campus Universitário de Cézeaux, 1 rue de la Chebarde, 63178, Aubière, França.

CLERMONT AUVERGNE INP

ÉCOLE DOCTORALE DES SCIENCES POUR L'INGÉNIEUR

LIMOS

(EN) This work was sponsored by a public grant overseen by the French National Research Agency as part of the *"Investissements d'Avenir"* through the IMobS^3 Laboratory of Excellence (ANR-10-LABX-0016) and the IDEX-ISITE initiative CAP 20-25 (ANR-16-IDEX-0001).

(FR) Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre du programme "Investissements d'Avenir" dans le cadre du Laboratoire d'Excellence IMobS^3 (ANR-10-LABX-0016) et de l'Initiative d'Excellence IDEX-ISITE CAP 20-25 (ANR-16-IDEX-0001).

(PT-BR) Este trabalho foi financiado por uma subvenção pública administrada pela Agência Nacional de Pesquisa Francesa como parte do programa *"Investissements d'Avenir"*, através do Laboratório de Excelência IMobS^3 (ANR-10-LABX-0016) e da iniciativa IDEX-ISITE CAP 20-25 (ANR-16-IDEX-0001).

Acknowledgements

I would like to thank my thesis supervisors Philippe Lacomme and Katyanne Farias for entrusting me with this thesis and also for their teaching, advice, support and guidance throughout these three years of my PhD. I have learned a lot from you!

I would also want to thank Manuel Iori for hosting me for two weeks at UNIMORE, Reggio Emilia, Italy, and to the research group *Recherche Opérationnelle et Décision* (ROD) and their *Action Jeunes Chercheurs* for providing partial funding for this stay.

I also extend my gratitude to the professors of the *Université Clermont Auvergne* (UCA) and the *Institut Supérieur d'Informatique, de Modélisation et de leurs Applications* (ISIMA) who entrusted me with their students for some classes (*travaux pratiques* and *dirigés*), which allowed me to develop teaching skills.

I thank the PhD students of the *Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes* (LIMOS) with whom I had the opportunity to exchange ideas throughout my thesis, especially those with whom I shared office B116 at ISIMA.

I thank Flavien Lucas, lecturer-researcher at the *Institut Mines-Télécom* (IMT) Nord Europe, for the collaboration and discussions on machine learning for inventory routing problems.

I thank the administrative and IT staff at LIMOS and ISIMA for hosting and helping me to solve problems and address my requests. I also thank the office managers of LIMOS and Labex IMobS^{3}, especially for helping me with the conference stays and administrative matters.

I also thank the reviewers and the members of my thesis committee for accepting the invitation and for the time dedicated to it.

Abstract

Inventory management and vehicle routing problems are logistic challenges that can significantly influence the efficiency and effectiveness of supply chain operations and should be well-coordinated and aligned. Handling both jointly is even more challenging when considering the number of customers to be served and the length of the time horizon. In the literature, this problem is known as the Inventory Routing Problem (IRP) and aims to find a minimum-cost solution that addresses both inventory and transportation problems simultaneously. The IRP was first introduced in 1983 by Bell et al. and have received a lot of attention from the OR community so far, which has introduced numerous extensions and provided datasets to favor research and fair comparisons.

In research, some gaps exist, and the IRP is not an exception. Most works in the literature so far assume that the fleet of vehicles used for the deliveries is homogeneous and that the costs associated with product storage and customer needs are constant and equal over the entire time horizon, which is not in accordance with a real scenario. Also, a single-item delivery per period is often considered by the formulation, which is clearly not cost-effective.

This thesis addresses the IRP and introduces a new variant that is closer to a real logistic scenario by incorporating a heterogeneous vehicle fleet, customer demands, and inventory holding costs that are period-dependent. Additionally, it considers that customers may prefer receiving products in batches rather than in single units. For that, a new set of instances is introduced to handle these new features. This novel variant, named the Heterogeneous Inventory Routing Problem with Batch Size (HIRP-BS), is studied using three approaches.

The first one is a mathematical formulation that extends a flow formulation initially designed to handle the HIRP-BS characteristics. New variables and constraints are then required to consider the new incorporated features. Not surprisingly, the formulation is not capable of handling large-scale instances and even the medium-scale ones are hard to solve in a timely manner.

The second method is an iterative algorithm which decomposes the original IRP into as many sub-problems as periods of time are considered. The idea is to solve the sub-problems in chronological order such that at each iteration (except for the first, which corresponds to the first period), it uses the solution obtained in the previous as a starting point for the current one. The changes are limited by an input parameter to accelerate convergence. The overall idea is that for a given period, the following iterations should require smart modification of the previous solutions of the partial problem already solved and that the number of changes should decrease once it approaches the end of the time horizon.

The third method is a split-based metaheuristic that decomposes a multi-period sequence of customers, called a giant tour, into routes that are assigned to a period and a vehicle type. The contribution leads to a new multi-period Split algorithm. It starts with the computation of the estimated quantities and periods for the replenishment, assuming the delivery operations at the latest possible moment. It allows the definition of a giant tour that is evaluated through a Split algorithm responsible for defining feasible solutions for the problem. Then, a local search mechanism dedicated to the routing problem takes advantage of classical route-based operators. Lastly, a post-optimization phase is considered, and slightly improve solution quality in terms of inventory and routing aspects based on a solution distance notion. Results are promising in terms of convergence and can provide valid upper bounds in a reasonable time even for the large-scale instances proposed.

Key-words : Logistics · Inventory management · Vehicle routing · Inventory Routing · Mixed Integer Linear Programming · Metaheuristic · Split algorithm

Résumé

Les problèmes de gestion de stock et de routage de véhicule sont des défis logistiques qui peuvent influencer de manière significative l'efficacité et l'efficience des opérations de la chaîne logistique et doivent être bien coordonnées et alignées. Les gérer conjointement est encore plus difficile lorsqu'on prend en compte le nombre de clients à servir et la durée de l'horizon de temps. Dans la littérature, ce problème est connu sous le nom de *Inventory Routing Problem* (IRP) et vise à trouver une solution de coût minimum qui traite les problèmes de stock et de transport simultanément. L'IRP a été introduit pour la première fois en 1983 par Bell et al. et a attiré jusqu'à présent l'attention de la communauté RO, qui a introduit de nombreuses extensions et fourni des données pour favoriser la recherche et les comparaisons justes.

En recherche, certaines lacunes existent, et l'IRP n'est pas une exception. La plupart des travaux existants supposent que la flotte de véhicules utilisée pour les livraisons est homogène et que les coûts associés au stockage des produits et aux besoins des clients sont constants et égaux tout au long de l'horizon, ce qui ne correspond pas à un scénario réel. De plus, la livraison d'un seul produit par période est souvent considérée, ce qui n'est pas rentable.

Cette thèse aborde l'IRP et introduit une nouvelle variante qui est plus proche d'un scénario logistique réel en incorporant une flotte de véhicules hétérogène, des demandes de clients et des coûts d'inventaire dépendants des périodes. De plus, on considère que les clients préfèrent recevoir des produits en lots plutôt qu'à l'unité. Pour cela, un nouveau jeu d'instances est introduit pour prendre en compte ces nouvelles caractéristiques. Cette variante, appelée *Heterogeneous Inventory Routing Problem with Batch Size* (HIRP-BS), est étudiée en utilisant trois approches.

La première est un modèle mathématique qui étend une formulation de flux existante pour incorporer les caractéristiques du HIRP-BS. De nouvelles variables et contraintes sont alors nécessaires pour cela. Il n'est pas surprenant que la formulation ne soit pas capable de résoudre les instances à grande échelle et que même celles à échelle moyenne soient difficiles à résoudre dans un temps raisonnable.

La deuxième méthode proposée est un algorithme itératif qui décompose l'IRP en autant de sous-problèmes que de périodes. Le but est de résoudre les sous-problèmes dans l'ordre chronologique et à chaque itération (à l'exception de la première, correspondant à la première période), d'utiliser la solution obtenue précédemment comme point de départ pour la période actuelle. Les changements sont limités par un paramètre d'entrée pour accélérer la convergence. L'idée générale est que pour une période donnée, les itérations suivantes devraient nécessiter de modifications intelligentes des solutions précédentes et que le nombre de changements devrait diminuer à mesure qu'on approche de la fin de l'horizon.

La troisième méthode est une métaheuristique basée sur un algorithme *split* qui décompose une séquence multi-période de clients, appelée tour géant, en routes qui sont attribués à une période et à un type de véhicule. L'algorithme débute par le calcul des quantités estimées et des périodes pour le réapprovisionnement, en supposant les opérations de livraison au dernier moment. Il permet la définition d'un tour géant qui est évalué à l'aide d'un algorithme *split* responsable pour définir des solutions réalisables pour le problème. Ensuite, un mécanisme de recherche locale dédié au problème de routage utilise les opérateurs classiques basés sur les routes. A la fin, une phase de post-optimisation est considérée, améliorant la qualité de la solution en termes de stock et transport, basée sur une notion de distance. Les résultats sont prometteurs en termes de convergence et peuvent fournir des bornes supérieures valides dans un délai raisonnable, même pour les instances à grande échelle.

Mots-clés : Logistique · Gestion de stock · Routage de véhicules · Inventory Routing · Mixed Integer Linear Programming · Métaheuristique · Algorithme split

Resumo

Os problemas de gestão de estoque e de roteamento de veículos são desafios logísticos que podem influenciar significativamente a eficiência e a eficácia das atividades da cadeia de suprimentos e precisam estar bem coordenadas e alinhadas. Tratar ambos simultaneamente é ainda mais desafiador quando se considera o número de clientes a serem servidos e o tamanho do horizonte temporal. Na literatura, este problema é conhecido como *Inventory Routing Problem* (IRP) e objetiva encontrar uma solução de custo mínimo considerando ambos os problemas de gestão de estoque e transporte concomitantemente. O IRP foi tratado pela primeira vez em 1983 por Bell et al. e recebeu muita atenção da comunidade de PO, o que motivou o estudo de novas extensões e conjuntos de dados que possam favorecer os estudos e estabelecer comparações justas.

Gaps existem na literatura e o IRP não é uma exceção. A maioria dos trabalhos da literatura assumem que a frota de veículos usada para as entregas é homogênea e que os custos associados com o armazenamento de produtos e as demandas dos clientes são constantes e iguais em todo o horizonte de tempo, o que não representa um cenário real. Também, as formulações existentes consideram a entrega de um único item por período, o que não é viável em termos de custos.

Esta tese aborda o IRP e introduz uma nova variante que se aproxima de um cenário logístico real, incorporando uma frota de veículos heterogênea e demandas dos clientes e custos de inventário dependentes dos períodos. Além disso, considera que os clientes preferem que as entregas sejam feitas em batches ao invés de um único item. Para isso, um novo conjunto de instâncias é apresentado para considerar essas novas características. Essa nova variante, chamada *Heterogeneous Inventory Routing Problem with Batch Size* (HIRP-BS), é estudada utilizando três abordagens.

A primeira é uma formulação matemática que estende uma formulação de fluxo da literatura a fim de tratar o HIRP-BS. Novas variáveis e restrições são necessárias para considerar as novas características. Sem surpresas, a formulação não é capaz de resolver instâncias de grande porte e mesmo as de médio porte são difíceis de serem resolvidas em tempo hábil.

A segunda é um algoritmo iterativo que decompõe o problema original em subproblemas de acordo com a quantidade de períodos considerados. O objetivo é de resolver os subproblemas em ordem cronológica e, a cada iteração (exceto para a primeira que corresponde ao primeiro período), utilizar a solução obtida na iteração precedente como ponto de partida para a atual. Mudanças na solução são limitadas por um parâmetro a fim de acelerar a convergência do método. Globalmente, para cada período, as iterações seguintes necessitam de mudanças inteligentes em relação às soluções anteriores que já resolvidas e que o numero de mudanças deve diminuir à medida em que se aproxima do final do horizonte temporal.

A terceira abordagem é uma metaheurística acoplada a um algoritmo de tipo *split* que decompõe uma sequência multi-período de clientes, chamada *giant tour*, em rotas que são atribuídas a um período e a um tipo de veículo. O algoritmo se inicia com a definição das quantidades estimadas a serem entregues e os respectivos períodos, considerando que estas acontecerão no período mais tarde possível. Isso permite a definição do *giant tour* que é avaliado através do algoritmo *split* e responsável por construir soluções viáveis para o problema. Em seguida, mecanismos de busca local dedicados à parte de roteamento considera operadores clássicos aplicados às rotas. Por fim, uma etapa de pós otimização permite melhorar a qualidade das soluções em termos de gestão de estoque e roteamento baseados na noção de distância. Os resultados são promissores em termos de convergência e podem definir Upper bounds validos em tempo hábil mesmo para as instâncias de grande porte.

Palavras-chave : Logística · Gestão de estoque · Roteamento de veículos · Inventory Routing · Mixed Integer Linear Programming · Metaheurística · Algoritmo split

Para Angélica, Aparecida e Jackson

Table of contents

Laboratory . . . 2
Funding . . . 2
Acknowledgements . . . 4
Abstract . . . 5
Résumé . . . 7
Resumo . . . 9

Introduction **19**

1 The Inventory Routing Problem **23**
1.1 Chapter introduction . . . 23
1.2 Problem definition . . . 24
1.2.1 Inventory management . . . 26
1.2.2 Routing problem . . . 37
1.3 Solution definition . . . 39
1.3.1 Distances matrix . . . 39
1.3.2 Routes scheduled . . . 41
1.3.3 Deliveries and inventory levels . . . 41
1.3.4 Solution cost . . . 43
1.4 Literature review . . . 44
1.4.1 Inventory Routing Problem . . . 44
1.4.2 Split algorithms . . . 47
1.5 Chapter conclusion . . . 57

2 Mathematical formulations **58**
2.1 Chapter introduction . . . 59
2.2 MILP formulation for the IRP . . . 59
2.2.1 Variables . . . 59
2.2.2 Objective function . . . 60
2.2.3 Constraints . . . 61
2.3 MILP formulation for the HIRP-BS . . . 63
2.3.1 Variables . . . 64
2.3.2 Objective function . . . 65
2.3.3 Constraints . . . 66
2.4 Instances . . . 68
2.4.1 IRP classical instances . . . 68
2.4.2 New HIRP-BS instances . . . 69
2.5 Experiments and results . . . 73
2.5.1 Literature instances . . . 73

2.5.2 New benchmark instances . . . 74
2.6 Chapter conclusion . . . 76

3 Iterative approach over periods 77
3.1 Chapter introduction . . . 77
3.2 Iterative algorithm . . . 78
3.3 Experiments . . . 82
3.3.1 Instances . . . 82
3.3.2 Results for the literature instances . . . 83
3.4 Chapter conclusion . . . 88

4 A Split-embedded Metaheuristic with a Post-optimization phase 89
4.1 Chapter introduction . . . 90
4.2 The SEMPO schema . . . 90
4.3 Giant tour generation and Split algorithm . . . 92
4.3.1 Generation of the giant tour . . . 94
4.3.2 Splitting the giant tour . . . 99
4.3.3 Shortest path retrieval . . . 107
4.4 Evolutionary Local Search algorithm . . . 108
4.4.1 Mutation . . . 110
4.4.2 Evaluation . . . 111
4.4.3 Local search . . . 111
4.5 Post-optimization . . . 119
4.5.1 Linearization constraints . . . 119
4.5.2 Distance definition . . . 119
4.6 Experiments and results . . . 120
4.6.1 Results for the literature instances . . . 121
4.6.2 Results for the new benchmark instances . . . 122
4.7 Chapter conclusion . . . 129

Conclusion 130

Publications 133

Extended resume (*in french*) 135
4.8 L'*Inventory Routing Problem* . . . 136
4.9 Formulations mathématiques . . . 137
4.9.1 l'IRP . . . 137
4.10 Métaheuristique basée sur un algorithme Split . . . 138
4.10.1 Algorithme Split . . . 139
4.10.2 Résultats . . . 140
4.11 Méthode itérative sur périodes . . . 140
4.11.1 Résultats . . . 141

A Instances examples 143
A.1 Classical IRP instance example . . . 143
A.1.1 How to read . . . 143
A.1.2 Example file . . . 144
A.2 New benchmark IRP instance example . . . 145

A.2.1 How to read 145
A.2.2 Example file 146

B IRP extended results **148**
B.1 Tables description 148
B.1.1 3-period instances 149
B.1.2 6-period instances 159

C HIRP-BS extended results **164**
C.1 Tables description 164
C.1.1 Small-scale instances 166
C.1.2 Medium-scale instances 170
C.1.3 Large-scale instances 179

D HIRP-BS convergence analysis **182**

Acronyms **185**

Bibliography **187**

List of tables

1.1 Customer inventory level calculation example . . . 27
1.2 Supplier inventory level calculation example . . . 29
1.3 X and Y localization coordinates . . . 40
1.4 Distance matrix . . . 40
1.5 Supplier inventory calculation . . . 42
1.6 Customer 1 inventory calculation . . . 42
1.7 Customer 2 inventory calculation . . . 42
1.8 Customer 3 inventory calculation . . . 42
1.9 Customer 4 inventory calculation . . . 42
1.10 Customer 5 inventory calculation . . . 42
1.11 Bibliography synthesis on the IRP papers contributions . . . 51
1.12 Overview on the IRP versions and approaches . . . 55

2.1 Sets, data and variables . . . 60
2.2 Description of sets, data and variables . . . 64
2.3 Literature instances characteristics . . . 69
2.4 Comparison of instance features . . . 69
2.5 Sets of instances for the HIRP-BS . . . 71
2.6 Results for the classical IRP set . . . 73
2.7 Results for the small HIRP-BS instances set . . . 74
2.8 Results for the medium HIRP-BS instances set . . . 75
2.9 Results for the large HIRP-BS instances set . . . 76

3.1 Results for the classical IRP instances . . . 83
3.2 Results for $\Delta = 50$ and 3 vehicles . . . 84
3.3 Results for $\Delta = 50$ and 3 vehicles . . . 84
3.4 Results for $\Delta = 50$ and 2 vehicles . . . 85
3.5 Results for $\Delta = 50$ and 2 vehicles . . . 85
3.6 Results for $|\mathcal{T}| = 3$ . . . 86
3.7 Results for $|\mathcal{T}| = 6$ . . . 86

4.1 Metaheuristic parameters . . . 121
4.2 Results on the high cost instances from Archetti et al. (2007) . . . 122
4.3 Results for the small-scale instances . . . 123
4.4 Gaps for the small-scale instances . . . 124
4.5 Results for the medium-scale instances . . . 125
4.6 Gaps for the medium-scale instances . . . 127
4.7 Results for the large-scale instances . . . 128
4.8 Gaps for the large-scale instances . . . 128

B.1 Literature instances characteristics . . . 148
B.2 IRP results for 3 periods, 1 vehicle . . . 149
B.3 IRP results for 3 periods, 2 vehicles . . . 151
B.4 IRP results for 3 periods, 3 vehicles . . . 153
B.5 IRP results for 3 periods, 4 vehicles . . . 155
B.6 IRP results for 3 periods, 5 vehicles . . . 157
B.7 IRP results for 6 periods, 1 vehicle . . . 159
B.8 IRP results for 6 periods, 2 vehicles . . . 160
B.9 IRP results for 6 periods, 3 vehicles . . . 161
B.10 IRP results for 6 periods, 4 vehicles . . . 162
B.11 IRP results for 6 periods, 5 vehicles . . . 163

C.1 (S) UB and metaheuristic comparison . . . 166
C.2 (S) RL and metaheuristic comparison . . . 167
C.3 (S) UB and metaheuristic time to target comparison . . . 168
C.4 (S) UB. RL and metaheuristic overview metrics . . . 169
C.5 (M) RL and metaheuristic comparison . . . 170
C.6 (M) UB and metaheuristic time to target comparison . . . 174
C.7 (M) RL and metaheuristic overview metrics . . . 177
C.8 (L) RL and metaheuristic comparison . . . 179
C.9 (L) UB and metaheuristic time to target comparison . . . 180
C.10 (L) LR and metaheuristic overview metrics . . . 181

List of figures

1.1 3-period IRP representation . . . 25
1.2 Inventory level evolution . . . 26
1.3 Customer delivered quantities . . . 27
1.4 Customer inventory level example . . . 28
1.5 Supplier production capacity . . . 29
1.6 Supplier production capacity . . . 30
1.7 Supplier inventory level update . . . 31
1.8 Anticipated inventory level . . . 32
1.9 Saving cost delivery anticipation . . . 32
1.10 ML replenishment policy . . . 33
1.11 OU replenishment policy . . . 34
1.12 ZIO replenishment policy . . . 35
1.13 BS replenishment policy . . . 36
1.14 Routing . . . 37
1.15 Alternation between search spaces (adapted from Prins et al. 2014) . . . 48

2.1 Example of a triangle inequality not verified . . . 73

3.1 Iterative heuristic idea . . . 78
3.2 The iterative heuristic search space evolution . . . 79

4.1 Schema of the SEMPO algorithm . . . 91
4.2 Split algorithm general idea . . . 93
4.3 Gain in routing cost when applying a λ probability . . . 96
4.4 Customers placement into Γ . . . 97
4.5 Labels propagation . . . 101
4.6 Addition of customer Γ_{j+3} in the route . . . 105
4.7 Addition of customer Γ_{j+3} in the route . . . 106
4.8 Delivery anticipation when adding an already visited customer . . . 106
4.9 Retrieving the critical path . . . 108
4.10 Walks on the solutions search space . . . 109
4.11 Swap mutation operator . . . 110
4.12 `2-OPT` inter routes . . . 112
4.13 `2-OPT` intra routes . . . 114
4.14 Insertion inter routes . . . 116
4.15 Insertion intra routes . . . 116
4.16 Post-optimization search schema . . . 120
4.17 Small-scale instances SEMPO convergence . . . 124

List of algorithms

1 Split for the CVRP . . . 49

2 Iterative algorithm . . . 81

3 Right-shift quantities . . . 95
4 Left-shift quantities . . . 95
5 Customers sequencing in the giant tour . . . 98
6 Split algorithm . . . 102
7 Evolutionary Local Search . . . 109
8 Mutation . . . 110
9 2-OPT inter routes . . . 113
10 2-OPT intra routes . . . 114
11 Insertion inter routes . . . 115
12 Insertion intra routes . . . 117
13 Global local search . . . 118

Introduction

Logistics operations encompasses the management of the flow of goods from its production to its final destination and must ensure efficient coordination of the operations. These operations comprise transportation and inventory management and must be very well coordinated to guarantee fluidity and sustainability of the supply chain. Handling both correctly is crucial to respond to the market demands and disruptions at the same time that enhance customer satisfaction and guarantee a competitive position among other companies.

Transportation management seeks for the choice of the best mode among rail, sea, air and road, aiming the minimization of the costs involved while guarantee, at the same time, reliability and safety. It also should deal with several problems that can represent real challenges, such as traffic congestion, infrastructure limitations, fuel costs and pollution, security and environmental aspects, for example. Addressing these scenarios require strategic planning and collaboration among the different people and industry involved. On the other hand, the inventory management considers the maintenance of the stock levels to meet the customers demands without stockouts while ensuring a satisfactory service quality. It faces some challenges regarding the demand prediction, the costs involved, the products life cycle, the loss prevention, the accurate stock balance to avoid stockout or overstock.

In order to try to handle both problems at the same time, Operations Research (OR) techniques can be applied. The problem of solving both transportation and inventory management simultaneously is named Inventory Routing Problem (IRP) in the literature. It drew the attention of many researchers over the last decades due to its complexity and practical relevance. It was introduced in 1983 by Bell et al. and as it happens for any other OR problems, the IRP objective is to find a minimum cost optimal solution, which in this case, is composed of both inventory and transportation costs. The IRP considers a set of customers and a supplier. The first has deterministic demands and the second a production capacity over a discrete time horizon. The inventory replenishment operations must be scheduled in order to avoid inventory disruption. To do so, a set of finite capacitated vehicles is available at each period and the routes containing the customers must be defined.

Due to its characteristics and the emergence for new features to be incorporated, several IRP variants have been studied, including, but not limited to:

- Safety stock, by imposing a minimum inventory level over the time horizon (Ramkumar et al. 2012 and Archetti et al. 2018)
- Maximum time limit for routes (Peres et al. 2017)
- Transshipment, with direct shipping between suppliers and customers or between customers (Coelho et al. 2012b, Azadeh et al. 2017 and Lefever et al. 2018)

- Perishable products, in which products deteriorate after a given time stored (Soysal et al. 2015, Azadeh et al. 2017 and Alvarez et al. 2020)
- Sustainable features, as summarized in Soysal et al. (2019)
- Green aspects considering hybrid vehicles (Gutierrez-Alcoba et al. 2023)
- Another variations as described in the surveys Andersson et al. (2010), Coelho et al. (2013) and Roldán et al. (2017).

Multiple are the authors that have proposed new extensions and methods to solve the IRP and its variants. Even if those are well tailored and introduce relevant features to the problem, we may notice that more work can be done to try to incorporate another characteristics that would turn the problem closer to what could be considered a real scenario. Obviously, it still remains a great challenge in OR problems.

In this sense and to guide the purpose of this thesis, the following questions can be raised:

→ *Which extra features can also be incorporated to the IRP?*

→ *Is the available data capable of handling additional characteristics?*

→ *Are the existing algorithms easy-to-adapt to handle new attributes?*

Besides addressing the classical IRP, this thesis introduces a novel extension of the IRP with intrinsic characteristics of real systems in order to provide answer elements these three precedent questions. Three features are added to the classical IRP: a heterogeneous and time-dependent vehicles fleet, delivery by batches and non-static demands and inventory holding costs. These three elements have either not been explored or have not received enough attention in the literature so far.

First, even though some previous works consider multiple vehicle types, most of the literature takes into account a homogeneous fleet of vehicles (as in, e.g., Desaulniers et al. 2016 and Bertazzi et al. 2019). Other authors, such as Coelho and Laporte (2014), model the IRP for a heterogeneous fleet of vehicles, but their method is tested only on instances with vehicles with identical capacity. Thus, benchmark instances for the IRP consider a homogeneous fleet of vehicles. Very recently, Skålnes et al. (2023) proposed a new set of instances, but also this set includes only homogeneous vehicles. In real-life, instead, the vehicle fleet is often heterogeneous, and the same set of vehicles is not always available over the whole time horizon as when the fleet is leased. For this reason, we consider an IRP with a heterogeneous fleet of vehicles whose availability varies over the time horizon.

Moreover, in real-world applications, products are often delivered to customers in batches rather than in single units. The delivery of a greater number of products, generating a positive inventory level, minimizes transportation costs since it reduces the number of customer visits to meet their demands over the time horizon. From an economic point of view, Wilson (1991) has shown that the average ordering costs are reduced with the growth of the number of items per order, which indicates an economic advantage when considering batch delivery. Accordingly, in the proposed new IRP variant, each customer has a predefined batch size, i.e., when a customer is visited, the quantity delivered must be a multiple of its batch value.

According to Archetti et al. (2007), two inventory policies exist: the Maximum Level (ML) and the Order-Up-to-Level (OU). The ML considers the fact that when a delivery is scheduled to a customer, any quantity that does not violate the existing and the maximum inventory level authorized can be chosen. However, in the OU policy, scheduling a customer to a route means fulfill its inventory level to its maximum capacity. Since these two policies does not correspond to the delivery by batches, the batch delivering proposed in this thesis can be seen as an emerging inventory policy for the IRP that has not been treated in the literature so far to the best of our knowledge.

In addition to the batch size per customer and the heterogeneous fleet of vehicles, and aiming to bring the problem closer to real scenarios, we consider non-static demands of customers and unit inventory costs varying over the time horizon. Consequently, and due to the emergence of this IRP variant and features, a new set of benchmark instances is presented to evaluate the efficiency of the proposed methods. These instances are challenging due to the existing characteristics that make the problem even more complex than the classical IRP, the large number of customers (up to 183) and the large size of the time horizon (up to 28 periods). Computational experiments and results are presented and discussed.

Similar to the classic version of the IRP, the objective is to satisfy all the customer demands at a minimum total cost of inventory holding and transportation, respecting the vehicle fleet capacity, the inventory level limits, the supplier production capacity and the fact that each customer is visited at most once per time period (no split delivery allowed). The problem is then named Heterogeneous Inventory Routing Problem with Batch Size (HIRP-BS).

To handle the HIRP-BS, we propose a Mixed Integer Linear Programming (MILP) flow formulation, a Split-Embedded Metaheuristic with a Post-Optimization phase (SEMPO) and an iterative algorithm over periods. The first extends the flow formulation presented by Archetti et al. (2014) for the classical IRP to incorporate and models the new features considered for the HIRP-BS. The proposed SEMPO algorithm is the core of this thesis and has mechanisms that provide high-quality solutions even for the new proposed large-scale instances with relatively low computational time. The SEMPO algorithm generates an initial solution using a giant tour Split approach, then it improves the solution by a local search procedure and finally attempts to further improve it by means of a post-optimization approach. This algorithm is derived from the Split-based algorithm originally developed for the Capacitated Arc Routing Problem (CARP), by Lacomme et al. (2001). Lastly, the iterative algorithm decomposes the original problem into subproblems that are solved sequentially according to the periods available.

The thesis is organized in four chapters. In Chapter 1, the IRP is presented and detailed according to both inventory management and transportation parts. The definition of a solution and its components is also described as well as a literature review on the main aspects of the IRP. Chapter 2 formalizes both IRP and HIRP-BS in mathematical terms by presenting the corresponding formulations including variables, objective function and constraints. Later, this Chapter presents two sets of instances: the classical one and the one introduced in this thesis to handle the new incorporated features. Results are also presented and discussed.

Chapter 3 presents an iterative algorithm which principle consists of defining small subproblems from the mathematical formulation of the IRP and solving each sequentially by adding at each iteration partial feasible solution to accelerate the convergence. Experiments and results are also introduced and discussed.

Lastly, Chapter 4 is the main Chapter of this thesis and presents the Split-based metaheuristic to solve both IRP and HIRP-BS problems. Initially, the split algorithm and its three steps to obtain a feasible solution are formalized. Then, seeking for diversification, the mutation and local search mechanisms are presented to optimize the transportation part of the problem and a post-optimization phase acting on both transportation and inventory parts is also introduced. Experiments and results are presented and discussed for both instances sets considered.

Chapter 1

The Inventory Routing Problem

Go to the Table of Contents ↺

Abstract

The Inventory Routing Problem stands as an integrated inventory management and transportation problem which considers a set of customers with deterministic demands and a supplier that must deliver a certain amount of products to each customer at each period of time using a fleet of vehicles available so that the demand of customers are met at a minimum total cost. This Chapter introduces the problem and illustrates how each part (transportation and inventory management) can be interpreted in order to obtain a minimum-cost solution value and respect a set of imposed constraints. A literature review is also presented to explore the problem and some of its variants as well as the resolution methods that can contribute to solve it efficiently.

Chapter content

1.1 Chapter introduction 23
1.2 Problem definition 24
1.2.1 Inventory management 26
1.2.2 Routing problem 37
1.3 Solution definition 39
1.3.1 Distances matrix 39
1.3.2 Routes scheduled 41
1.3.3 Deliveries and inventory levels 41
1.3.4 Solution cost 43
1.4 Literature review 44
1.4.1 Inventory Routing Problem 44
1.4.2 Split algorithms 47
1.5 Chapter conclusion 57

1.1 Chapter introduction

The Inventory Routing Problem (IRP) is a largely studied problem in the Operations Research field and incorporates an inventory management and a routing problem. Both consider a supplier responsible for the delivery of a unique or multiple types of products to a set of customers that need to be served within a finite discrete time horizon. The pioneers authors to address the IRP are authors Bell et al. (1983) that have presented the case of a gases industry that have a real time optimizer to solve the problem by a mathematical formulation with a sophisticated Lagrangian relaxation (Fisher 1981) and has permitted the industry to save costs in their production system.

In this chapter, the IRP is presented from the two different points of view and the aspects related to each one are detailed and justified with the existing literature. Also, the details of a feasible solution including costs and inventory levels calculation and routes definition, for example, are also presented. Lastly, the literature related works on the problem closes the Chapter.

In Section 1.2, the inventory management part is introduced in Subsection 1.2.1 and corresponds to the inventory management problem, partially responsible for the delivery operations (since there is also a dependance on the routing associated problem) and the customer replenishment over the periods of time available. An analysis is provided for the customer and the supplier, as well as the different inventory policies considered in the literature. The routing problem is presented in Subsection 1.2.2. It presents how the vehicles fleet is composed as well as other aspects related to the customers visitation all over the periods.

In Section 1.3, the elements that compose a feasible solution for the IRP are detailed with a numerical example and include the definition of the distance matrix, the routes that are assigned to the available vehicles, the delivery operations that define the inventory level for each customer and supplier as well as the costs involved with these operations. Lastly, in Sections 1.4, relevant literature on the IRP that have treated the problem and its variants and the existing algorithms to solve the problem are presented and confirm the interest of its resolution by incorporating interesting features and characteristics. Also, the approaches used to solve them and obtain feasible upper bounds are presented and discussed. This bibliography may also confirm the interest of incorporating other characteristics and present more instances that help the problem to get closer to a real scenario, which is one of the purposes of this thesis.

1.2 Problem definition

As previously introduced, the IRP has routing and inventory components that work together. Figure 1.1 presents a simple schema illustrating the IRP in terms of these two elements:

- For the routing part, the arcs are represented by three colors corresponding to different periods of time, leading to a three-period IRP. Customers are represented by vertices (circles), while the supplier is depicted at the center as a black square. From the supplier, all vehicle routes originate to deliver products to the customers, returning once the scheduled customers have been served. Each route is assigned to a single vehicle, which serves a subset of customers. Note that the same customer can be visited at most once per period but may be visited in all the periods considered.

- At the bottom right of the schema (Figure 1.1), a detailed view focuses on a customer to illustrate how its inventory level evolves when a delivery operation is performed. The first orange bar represents the inventory level at the beginning of the period, before the arrival of the vehicle to the customer or the supplier. The green bar corresponds to the delivery amount, while the red bar represents the customers consumption (demand). Note that for this customer, deliveries are performed only in periods 1 and 2 since in period 3 the inventory level is enough.

Note that when a vehicle is assigned to a route, it carries all the products that must be delivered to the customers scheduled on his path and the inventory levels are supposed to be respected since the decision on when and how much to deliver is stablished before the vehicle departure from the warehouse (supplier).

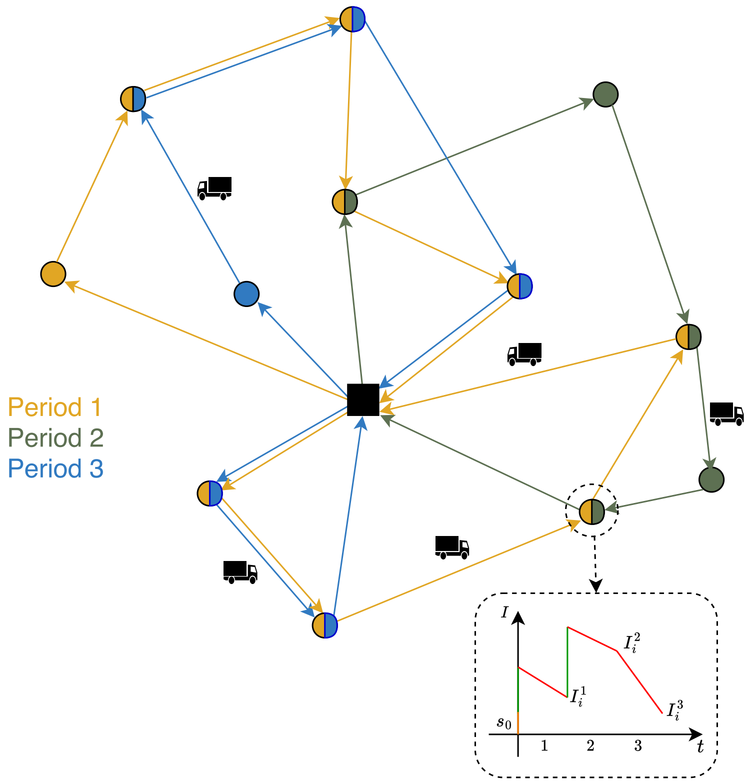


Figure 1.1: 3-period IRP representation

Problem assumptions

Despite all the existing IRP variants, some assumptions on the problem may never change according to each variant. These are fundamental to understand the basis of the problem and how each transportation and inventory management associated problems work together. Below, the most important ones are cited.

- **Split delivery is not allowed.** Each customer is visited only once per period and at most the number of periods over the time horizon.
- **Delivery and consumption.** The delivery occurs only at the beginning of a period and the consumption is linear considering a time interval.
- **Inventory levels calculation.** Inventory levels are calculated at the end of the period once all the operations have been concluded. These operations include the replenishment (delivery of an amount of products), the demand deduction and inventory level from the immediately previous period.

- **Vehicles capacity.** Vehicles that perform the deliveries have a finite capacity and can leave the depot only if there is at least one customer to be served, i.e., with at least one product unit inside.
- **Availability of products to be delivered.** The supplier has always enough products available at each period to supply all the customers once a demand exists.
- **Maximum and minimum inventory levels.** Customers have minimum and maximum inventory levels to be respected. Usually, the minimum is equal to zero. Note that for the supplier, only the minimum level exists and is always equal to zero.
- **Stock disruption.** In any case, a stock disruption is not authorized.
- **Routes feasibility.** A valid route always starts and ends at the depot. A sub-tour (a route not starting or finishing at the depot) must be eliminated by special routing constraints. Also, each route is assigned to only one vehicle.

1.2.1 Inventory management

The inventory levels over the time horizon takes into account the initial level s_i for the fictitious period 0, the amount of product q_i^t delivered at a each time period, the customers demands d_i^t for each period t as well as the supplier production capacity expressed by r^t. Each customer has a maximum inventory level allowed U_i. The inventory levels are defined by I_i^t for a given customer i or supplier in period t and corresponds to the inventory level at the end of period t once the delivery (q_i^t) and the demand deduction (d_i^t) have been performed as shown in Figure 1.2. Thus, $0 \leq I_i^t \leq U_i$.

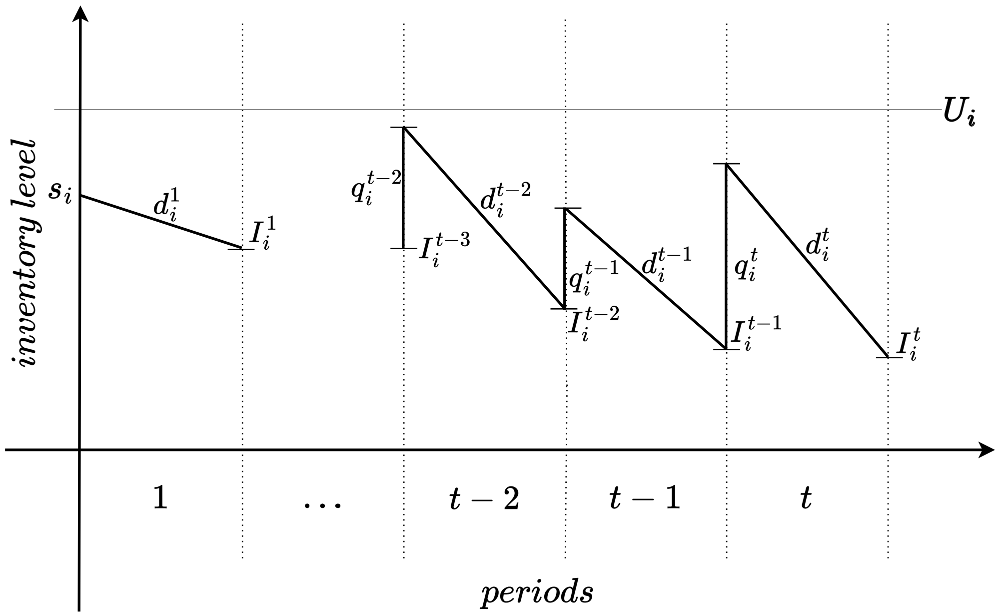


Figure 1.2: Inventory level evolution

Note that the inventory levels can be analyzed from the supplier and the customer point of views and their calculation is not the same. The next subsections explains both calculations.

Customer and supplier points of view

1. Customer inventory level

   In Figure 1.2, for a customer i, the inventory level evolution over t periods is represented. The first delivery operation is performed at period $t-2$ with the delivery of q_i^{t-2} unities of products and the inventory level I_i^{t-2} is calculated at the end of this period since the problem considers discret periods of time which means that the order of the operations (consumption/delivery) performed between the end of a period and its precedent does not influence the inventory calculation.

   First, if we take a look only at the quantities that are delivered from the supplier, we can interpret them as a linear function that depends only on a given customer i and a period t. This scenario is represented on Figure 1.3 for a time horizon equal to 7 periods and the data comes from the example presented in Table 1.1 for a given customer i with an initial inventory level $s_0 = 30$. Column t corresponds to the period, q_i^t to the quantity delivered, d_i^t to the demand and I_i^t to the inventory level calculated at the end of period t.

t	q_i^t	d_i^t	I_i^t
0	-	-	30
1	30	25	30 + 30 - 25 = 35
2	10	15	35 + 10 - 15 = 30
3	0	20	30 + 0 - 20 = 10
4	50	20	10 + 50 - 20 = 40
5	20	10	40 + 20 - 10 = 50
6	10	5	50 + 10 - 5 = 55
7	0	50	55 + 0 - 50 = 5

Table 1.1: Customer inventory level calculation example

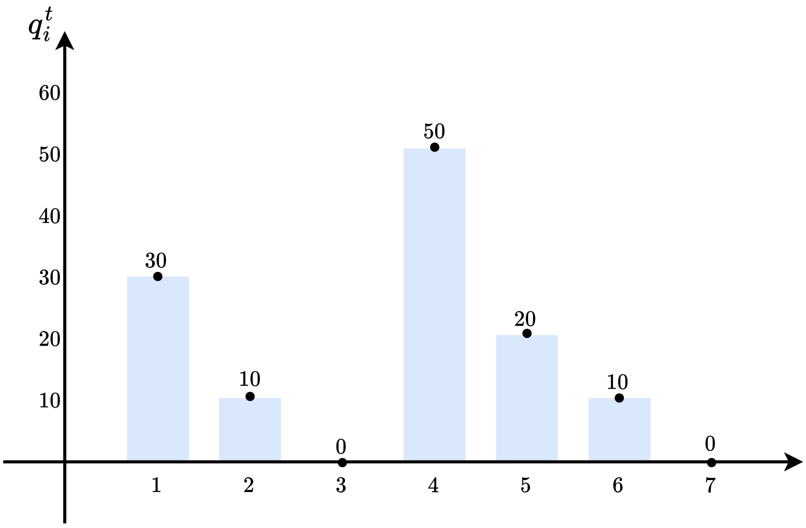


Figure 1.3: Customer delivered quantities

From these quantities delivered, the inventory level for a given customer i and period t can be defined by Equation 1.1. Note that in this case, the level is dependent on the previous ones, the quantities and the demands. Figure 1.4 illustrates the evolution of the inventory level. Green bar represents the quantity delivered and the red bar only the demand. Dotted line indicates the inventory level once the delivery and consumption are done.

$$f(i,t) = s_i + \sum_{p=1}^{t} (q_i^p - d_i^p) \tag{1.1}$$

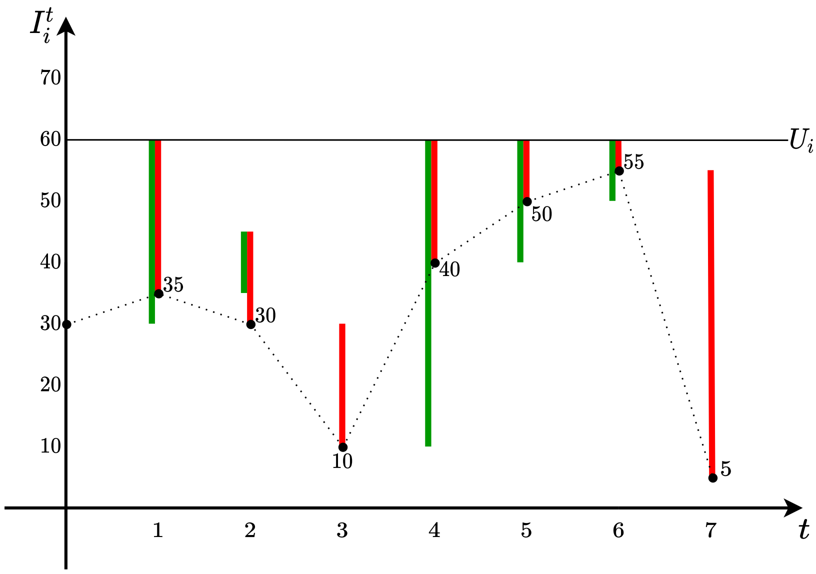


Figure 1.4: Customer inventory level example

Note that the assumption is that the inventory level constraints are applicable considering the end of a given time period as previously stated, i.e., once the quantities have been delivered and the demand has been consumed. However, it must also take into account the remaining space available to define the quantity to be replenished. This remaining space is calculated by subtracting from the customer maximum inventory level allowed the inventory level at the beginning of the period. That is why in Figure 1.4 we see that in periods 1, 4, 5 and 6, all the remaining space is fulfilled with products but never surpass the maximum limit imposed by U_i.

2. Supplier inventory level

   Now, from the supplier point of view, the production capacity r^t for a given period t is illustrated by Figure 1.5 with data from Table 1.2 and considering an initial inventory level $s_0 = 600$. This capacity is seen as the amount of products that the supplier makes available for the set of customers at each period. The non used products are cumulated for the subsequent periods. Column t corresponds to the period, r^t to the supplier production capacity, $\sum q_i^t$ to the total amount of products delivered to all the customers in period t and I_0^t to the supplier inventory level also calculated at the end of period t.

t	r^t	$\sum q_i^t$	I_0^t
0	-	-	600
1	200	450	600 + 200 - 450 = 350
2	400	300	350 + 400 - 300 = 450
3	300	350	450 + 300 - 350 = 400
4	500	550	400 + 500 - 550 = 350
5	550	200	350 + 550 - 200 = 700
6	150	300	700 + 150 - 300 = 550
7	350	450	550 + 350 - 450 = 450

Table 1.2: Supplier inventory level calculation example

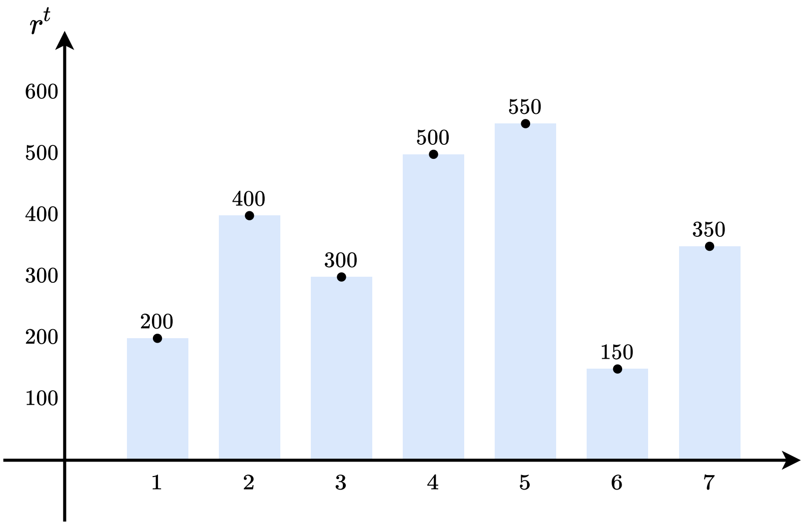


Figure 1.5: Supplier production capacity

Similarly to the customers, the supplier inventory level is defined by Equation 1.2 and takes into account the initial inventory level s_0, the total quantity of products produced r^t at each period t and the total amount q_i^t that is delivered to the $|\mathcal{N}|$ customers. Figure 1.6 shows the inventory level evolution over the 7 periods of time.

$$f(0,t) = s_0 + \sum_{p=1}^{t} \left(r^t - \sum_{i=1}^{|\mathcal{N}|} q_i^p \right) \tag{1.2}$$

Note that the supplier does not have a maximum capacity as it occurs for the customers. We consider that his production capacity is enough to meet all the customers demands over the time horizon.

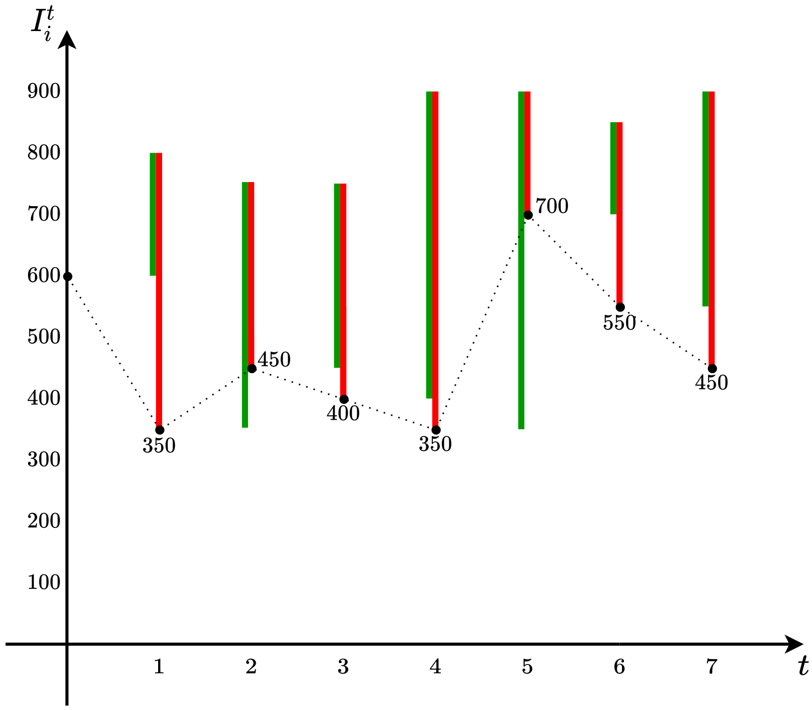


Figure 1.6: Supplier production capacity

□

In contrast to the classical VRP, in which the quantities to be delivered are given as problem data, in the HIRP-BS, as for the IRP, the quantities to be delivered are variables of the problem and must be specified in the solution. The total volume of deliverable products in a period $t \in \mathcal{T}$ is constrained partly by the volume r^t produced by the supplier and the total capacity of the vehicle fleet, which relies on the number of available vehicles and their respective capacities.

Therefore, the sum of the customer demands for a given period $t \in \mathcal{T}$ may exceed the delivery capacity of the vehicle fleet (C) or exceeds the volume of products made available by the supplier (r^t). Consequently, some demands could not be met when needed and must be delivered during the previous periods. It may also be advantageous to anticipate the deliveries to minimize the transport and total costs, even if this incurs an additional storage cost.

An anticipated delivery incurs an inventory level greater than zero and, consequently, a storage cost. Additionally, each customer $i \in \mathcal{N}$ has a maximum inventory level allowed denoted by U_i. One challenge of the IRP is to determine which quantities of products should be delivered to each customer and in which period, aiming to meet the customers demands for each period, taking into account the capacity of the vehicle fleet, the volume r^t produced by the supplier and the customers inventory capacity.

At each period $t \in \mathcal{T}$, r^t products arrive at the supplier inventory and the evolution of their stock levels depend on the quantities delivered to the customers as presented in Figure 1.7 and illustrates the delivery of q_1^1 and q_2^1 unities of products to customers 1 and 2 in period 2 and this amount is deducted from the supplier inventory level and also considers the production of r^2 unities.

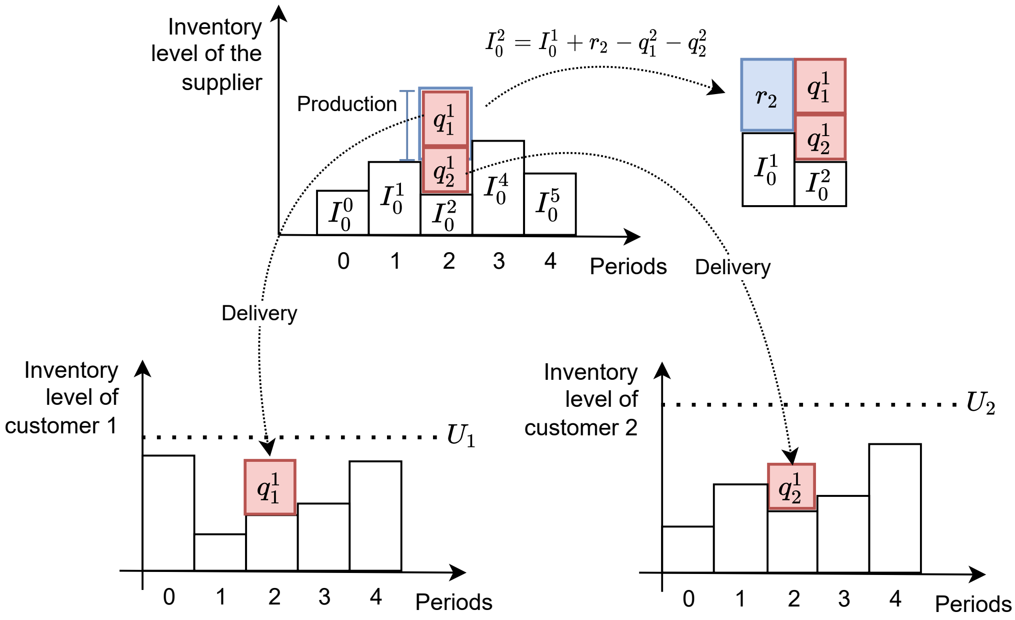


Figure 1.7: Supplier inventory level update

Another challenge of the IRP resolution refers to the delivery of products to the customers as late as possible so that the demands are met and the inventory holding costs are the minimum possible. This scenario may exceed the total vehicle fleet capacity for a given period and anticipating a part of the deliveries is necessary in such a way that the vehicle fleet capacity is respected. However, it leads to an increase in the inventory costs over the time horizon. Figure 1.8 illustrates this case by showing that in period 6, the total delivery quantity exceeds the vehicle fleet C and then an amount of products is anticipated to periods 3 and 5. Consequently, the inventory costs observed increase.

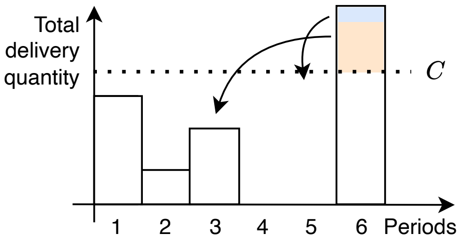
Another case in which it is advantageous to anticipate deliveries, even if resource capabilities are respected, is when the total cost is minimized due to the gain on the transportation cost despite the increase in the inventory costs because the number of visits to one customer is reduced and the transportation costs are lower than the inventory costs as shown in Figure 1.9.

The case described in Figure 1.9 illustrates a delivery to a customer i that is brought forward from period 5 to 3 considering that all constraints are respected. This increases the inventory level of customer i from period 3 onwards and, consequently, the total inventory cost. However, customer i has already visited in period 3 and adding a supplementary quantity to this period does not increase the corresponding transportation cost. In addition, removing customer i from period 5 reduces the corresponding transportation cost, in such a way that the total cost is minimized.

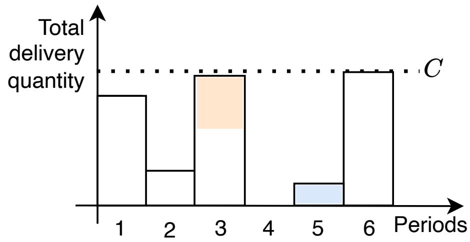
Inventory policies

When regarding the quantities to be delivered, two policies are studied in the literature: Maximum Level (ML) and Order-Up to level (OU). Both are described below.

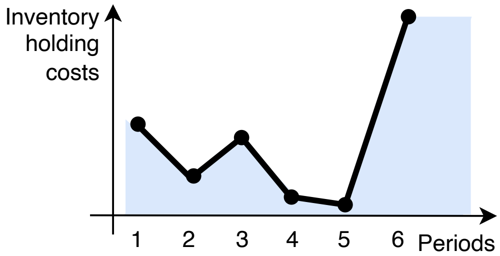
For the sake of comprehension, we consider $x_{i,j}^t$ an indicator that takes the value 1 if customer i comes just before another customer j in a route performed at period t.



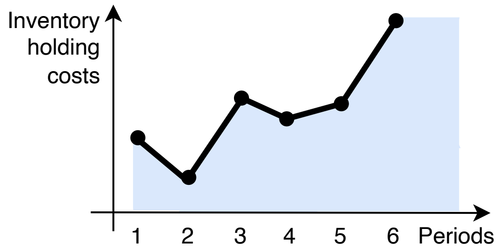
(a) Latest deliveries



(b) Anticipated deliveries

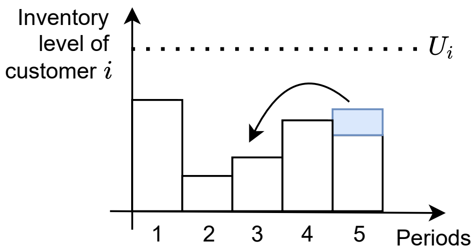


(c) Latest inventory level

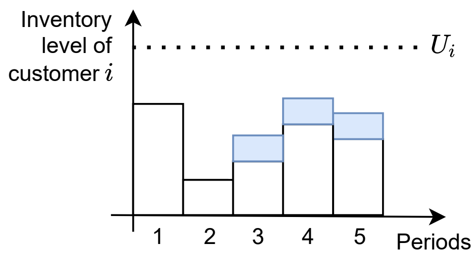


(d) Transportation total cost

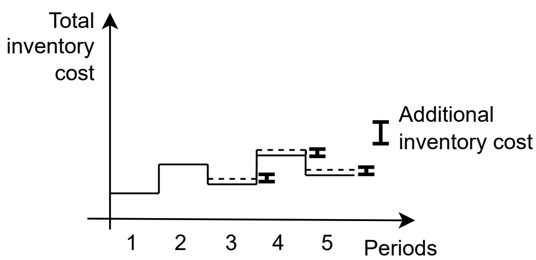
Figure 1.8: Anticipated inventory level



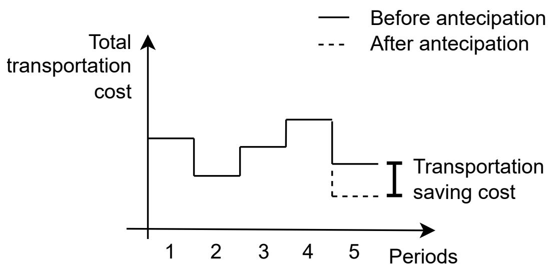
(a) Before anticipation



(b) After anticipation



(c) Inventory total cost



(d) Transportation total cost

Figure 1.9: Saving cost delivery anticipation

(i) Maximum Level (ML)

The Maximum Level (ML) inventory policy considers that a customer can receive any quantity of a given product at any moment since the inventory limitations, i.e., minimum and maximum levels are respected as shown in Figure 1.10.

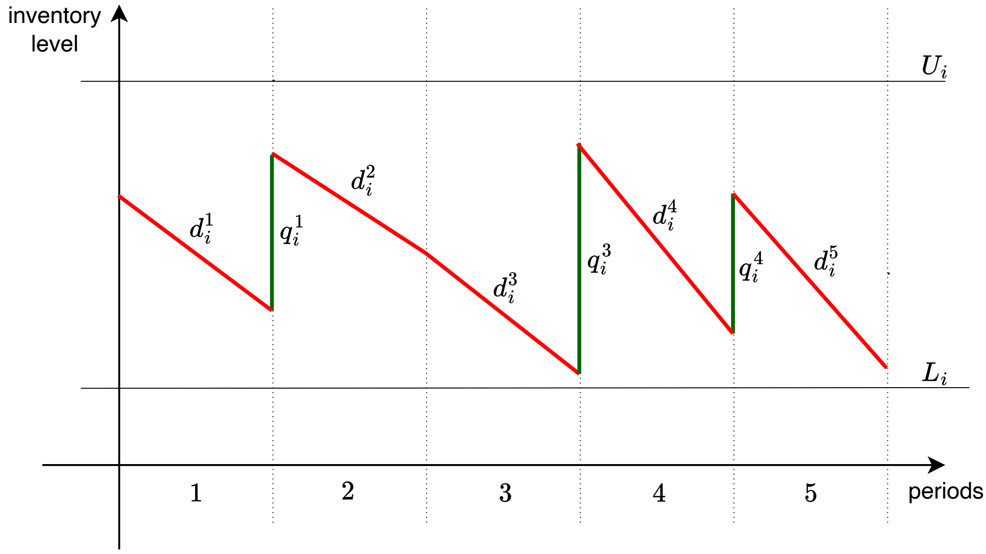


Figure 1.10: ML replenishment policy

In Figure 1.10, the red bars represent the consumption of products (demand) and the green ones refer to the quantity that is delivered. Note that once the demands are deducted and the products are added to the customer inventory, the inventory level never surpass the limits imposed and correspond to any value between both levels L_i (min) and U_i (max).

In mathematical terms, we have the following constraints (adapted from Archetti et al. (2007)):

$$q_i^t \geq L_i - I_i^{t-1} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.3}$$

$$q_i^t \leq U_i - I_i^{t-1} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.4}$$

Constraints 1.3 ensure that the quantity to be delivered must respect the minimum inventory level (L_i) and the same is observed for constraints 1.4 concerning the maximum level (U_i). Both ensure that the quantity (q_i^t) also depends on the remaining space available taking into account the previous inventory level (I_i^{t-1}).

(ii) Order-up-to Level (OU)

The Order-up-to level (OU) policy states that once a customer is visited at any time period, the amount of product that is supposed to be delivered corresponds to the exact number of products to fill the inventory to the highest level possible (U_i) as illustrated by Figure 1.11.

Note that in Figure 1.11, differently from the ML policy, once a delivery is made, the customer inventory level is filled to its maximum capacity (max).

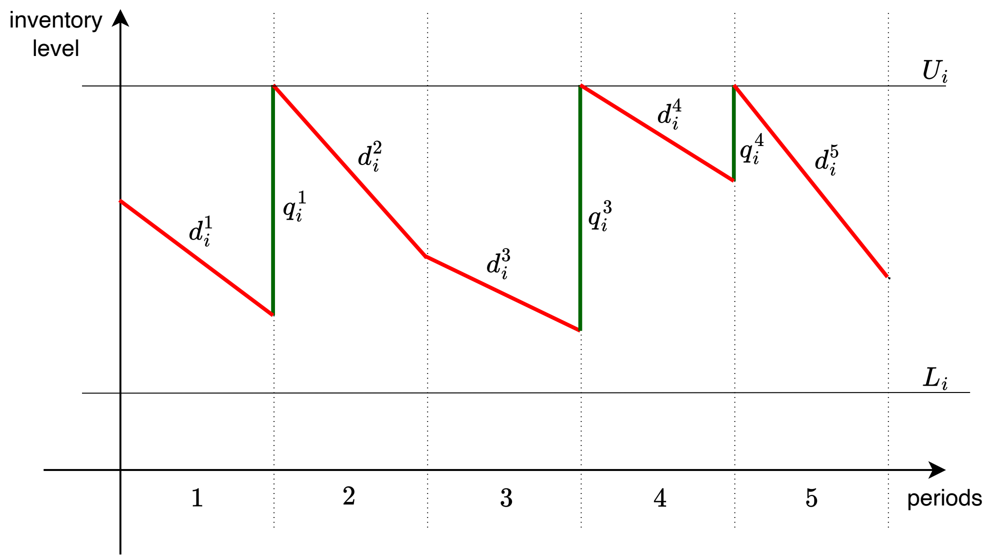


Figure 1.11: OU replenishment policy

Also, the following constraints are are used to handle this policy (adapted from Archetti et al. (2007)):

$$q_i^t \geq U_i z_i^t - I_i^{t-1} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.5}$$

$$q_i^t \leq U_i - I_i^{t-1} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.6}$$

$$q_i^t \leq U_i z_i^t \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.7}$$

$$q_i^t \geq L_i - I_i^{t-1} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.8}$$

z_i^t is a binary variable equal to 1 is customer i is visited in period t and 0 otherwise. These three constraints guarantee that when a customer i is visited ($z_i^t = 1$), the quantity delivered is equal to the remaining space available considering the customer maximum inventory level allowed (U_i) and the precedent inventory level (I_i^t). Otherwise, if the customer is not visited in a given period ($z_i^t = 0$), then nothing is delivered. What follows explain these two possibilities.

$$\textbf{If } z_i^t = 1 \textbf{ then } \begin{cases} q_i^t \geq U_i - I_i^{t-1} \textbf{ and } q_i^t \leq U_i - I_i^{t-1} \Rightarrow q_i^t = U_i - I_i^{t-1} \\ q_i^t \leq U_i (\text{redundant}) \end{cases} \tag{1.9}$$

$$\textbf{If } z_i^t = 0 \textbf{ then } \begin{cases} q_i^t \leq 0 \Rightarrow q_i^t = 0 \text{ (because } q_i^t \geq 0 \text{ always)} \\ q_i^t \geq -I_i^{t-1} \textbf{ and } q_i^t \leq U_i - I_i^{t-1} \text{ (redundant)} \end{cases} \tag{1.10}$$

(iii) Zero-Inventory-Ordering (ZIO)

The Zero-Inventory-Ordering (ZIO) policy consists of allowing delivery operations from the supplier to the customer only and only if the customer inventory level is equal to zero. Figure 1.12 illustrates this replenishment policy. Two deliveries are scheduled in periods 2 and 3 and those happen only if the previous inventory level is equal to zero.

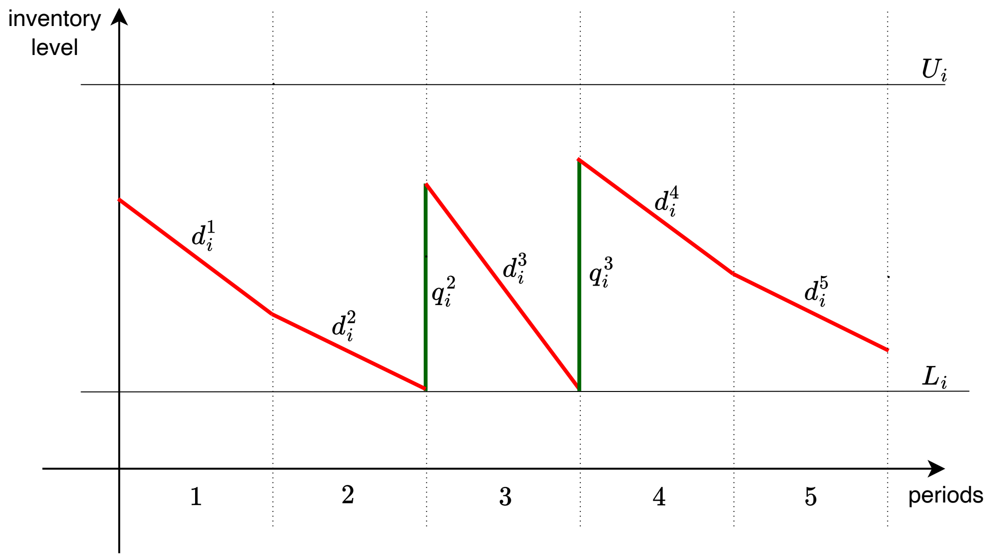


Figure 1.12: ZIO replenishment policy

Diabat et al. (2024) have classified this policy as customer-oriented since it aims to reduce the customers inventory costs by keeping its level as low as possible and also reduces the number of visits since a visit is made only if necessary. These same authors have proposed a mathematical formulation to solve the IRP considering this policy and three constraints have been added to the classical ones to take into account this policy. An adapted version reducing at most the number of variables for the sake of comprehension is presented below.

$$I_i^{t-1} \leq U_i(1 - z_i^t) \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.11}$$

$$I_i^{t-1} \geq 1 - z_i^t \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.12}$$

z_i^t consists of a binary variable equal to 1 is customer i is visited in period t and 0 otherwise. Both Constraints 1.11 and 1.12 ensure that if a customer is visited ($z_i^t = 1$), then the inventory level for this customer is equal to 0 in the previous period and that the inventory level is comprised in the interval [1,U_i] otherwise ($z_i^t = 0$), as follows.

$$\textbf{If } z_i^t = 1 \textbf{ then } I_i^{t-1} \leq 0 \textbf{ and } I_i^{t-1} \geq 0 \Rightarrow I_i^{t-1} = 0 \tag{1.13}$$

$$\textbf{If } z_i^t = 0 \textbf{ then } I_i^{t-1} \leq U_i \textbf{ and } I_i^{t-1} \geq 1 \Rightarrow I_i^{t-1} \in [1, U_i] \tag{1.14}$$

Note that the minimum inventory level L_i is not taken into account in these equations since for all the instances considered, this minimum level is equal to 0 and since the inventory level cannot be lower than zero according to the variable domain, it is redundant to keep the constraints related to L_i.

As stressed by these same authors (Diabat et al. (2024)), this policy inconvenient is that the initial inventory level of any customer must correspond to the sum of the demands from period 0 to any of the periods considered. It can be easily proved: if we consider an horizon from 0 to any t, then if $s_0 \neq \sum_0^t d_i^t$, no feasible solution exists since the inventory level will never be equal to 0 and then no delivery will be scheduled and, consequently, a stock disruption will be observed, leading to an unfeasible solution.

(iv) Batch size (BS)

The majority of the IRP formulations considers the fact that one customer can be visited in a given period to receive only one unit of a given product. This situation arises because there are no limits on the values of the q_i^t variables to impose a minimum authorized value. To tackle this situation, this thesis addresses a new inventory policy named Batch Size (BS) which consists in considering for each customer a batch size that must be considered for the deliveries and the total number of products per period must be multiple of this predefined value. This scenario is illustrated in Figure 1.13. Note that in practice, this policy can be helpful to facilitate the management and the placement of the products into the warehouses and customers storage places.

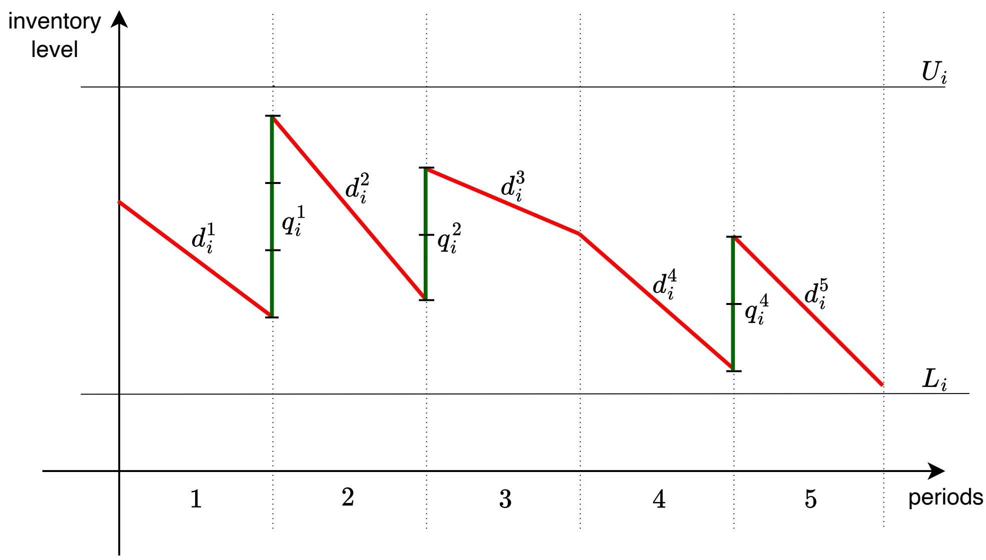


Figure 1.13: BS replenishment policy

In Figure 1.13, for a given customer, each batch is represented by the vertical green line separated by the black dashes. In this case, three batches were delivered in period 1, followed by two in period 2 and another two in period 4.

To handle this policy and impose the quantity to be multiple of the customer batch size, only one constraint is required. It is expressed by the equality given in Equation 1.15. Variable b_i^t represents the number of batches delivered to customer i in period t and ℓ_i the customer fixed batch size. Additional constraints 1.16 and 1.17 from the ML policy must also be considered to manage the available space into the customer inventory.

$$q_i^t = b_i^t \ell_i \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.15}$$
$$q_i^t \geq L_i - I_i^{t-1} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.16}$$
$$q_i^t \leq U_i - I_i^{t-1} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{1.17}$$

The BS policy is closer to the ML in such a way that the quantity to be delivered can be any if two constraints are considered:

(a) the quantity must be multiple of the customer batch size

(b) the quantity must be under or equal the remaining space available at a customer inventory

It differs from the other policies in such a way that the amount delivered may not fulfill the customer inventory at its maximum authorized level as it occurs in the OU policy and that a replenishment is totally possible even if the customer previous inventory level is above 0 as it occurs in the ZIO policy.

1.2.2 Routing problem

The routing problem relies on the definition of the routes to be performed by a set of vehicles assigned to a given period of time in order to execute the delivery operations to customers. It consists in defining the customers to be visited by a vehicle in which the capacity is finite. The vehicles must leave the depot (supplier), visit the customers and return to the depot once all the scheduled customers have been treated. Figure 1.14 illustrates the routing problem for three periods of time and four customers (A, B, C, D) to be treated before and after the routes schedule.

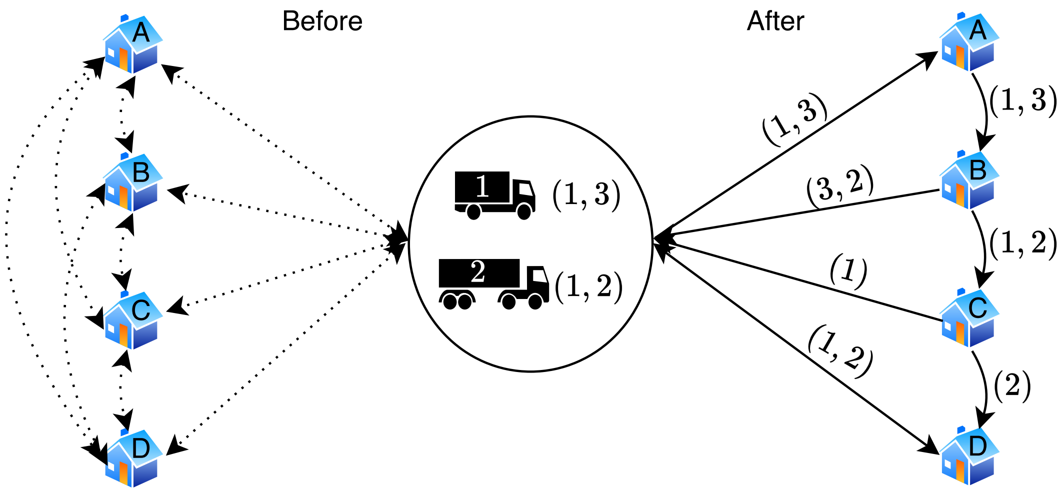


Figure 1.14: Routing

On the left side of Figure 1.14, the complete graph is available to define the routes. Two vehicles $(1, 2)$ are available for each of the three periods. On the right side, the customers are assigned to a route and each route is allocated to a single vehicle. Numbers in parenthesis represents the affectation period. The routes are the following (0 stands for the supplier):

- Period 1
  - Vehicle type assigned: 1
  - Visitation sequence: $0, A, B, C, 0$

  - Vehicle type assigned: 2
  - Visitation sequence: $0, D, 0$
- Period 2
  - Vehicle type assigned: 2
  - Visitation sequence: $0, B, C, D, 0$
- Period 3
  - Vehicle type assigned: 1
  - Visitation sequence: $0, A, B, 0$

As in other routing problems such as the Vehicle Routing Problem (Eksioglu et al. (2009); Baker and Ayechew (2003)), the Heterogeneous Vehicle Routing Problem (Gendreau et al. (1999); Molina et al. (2020)), Capacitated Vehicle Routing Problem (Uchoa et al. (2017); Toth and Vigo (2002)), sub-tours constraints have to be incorporated to the IRP to guarantee that a route is feasible.

Vehicle fleet types

In routing problems, as previously seen, the fleet of vehicles can be homogeneous or heterogeneous. In classical versions of these problems, the fleet is in the majority of the cases considered as homogeneous but in real scenarios, it is often heterogeneous and can increase considerably the complexity of the problem. Here, the fleet composition is discussed and examples are given for each case.

(i) Homogeneous

The majority of papers that treated the IRP and variations consider a homogeneous vehicle fleet in their algorithms. Archetti et al. (2007) when introduced a benchmarking set of instances have considered only a single homogeneous vehicle per period. Later, Coelho et al. (2012a) have used the same set of instances to treat the multi vehicle case up to 5 homogeneous vehicles per period. In Skålnes et al. (2023), a new set of instances is introduced for the IRP but also only considers a homogeneous vehicle fleet even if three vehicle dimensions are presented, each instance is considered to be homogeneous and uses only one from the three vehicle types available.

(ii) Heterogeneous

As Hoff et al. (2010), the fleet of vehicles is rarely homogeneous and can vary on physical, compatibility and costs aspects. The heterogeneous vehicle fleet was introduced by Golden et al. (1984) and named Fleet Size and Mix Vehicle Routing Problem (FSMVRP). It consists in defining the optimal size of the fleet considering a fixed cost based on the leasing rather than the acquisition of the vehicle and assumes that the vehicles costs and capacity are not always the same.

Goel and Gruhn (2008) have studied the General Vehicle Routing Problem (GVRP) which consists, among others, in considering time windows, compatibility and route restrictions constraints in order to take the problem closer to a real life scenario. A survey on the almost thirty years of the HVRP is presented in Koç et al. (2016).

1.3 Solution definition

Finding a feasible solution for the IRP means defining, for each time period, the visits to the customers, the number of products delivered, the vehicles assigned to the routes, and the customer visit sequences for each route, taking into account inventory management and transportation constraints related to both customers and supplier.

Let's consider for the next Subsections the instance file `abs1n5` below which contains 5 customers and the supplier and was extracted from the 3-period and 2-vehicle high cost instances set from Archetti et al. (2012). The data example is issued from this instance and from its optimal solution.

File 1.1: Literature instance `abs1n5`

```
6  3  289
   1        154.0       417.0            510             193          .30
   2        172.0       334.0   130   195       0    65          .23
   3        267.0        87.0    70   105       0    35          .32
   4        148.0       433.0    58   116       0    58          .33
   5        355.0       444.0    48    72       0    24          .23
   6         38.0       152.0    11    22       0    11          .18
```

In the first line, number `6` corresponds to the number of customers and suppliers (5 customers, 1 supplier), `3` to the number of periods and `289` to the transportation capacity per period. The second line provides information on the supplier, in that order: index, `X` and `Y` location coordinates, initial inventory level, production capacity per period and the inventory cost per unit of product. From the third line and on, the information is, in that order: index, `X` and `Y` location coordinates, initial inventory level, maximum and minimum inventory level allowed per period, demand per period and inventory cost per unit of product.

1.3.1 Distances matrix

The customers and supplier `X` and `Y` coordinates are given and the distance matrix can be obtained by applying the Euclidian distance equation as stated by Theorem 1.3.1.

Theorem 1.3.1 (Euclidian distance) *The Euclidian distance between two pairs of coordinates* $\{(x_1, y_1), (x_2, y_2)\}$ *is given by*

$$EucDist((x_1, y_1), (x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1.18}$$

and corresponds to the length of the segment that relies both points in a two-dimensional plane and can be interpreted as the shortest path between them.

□

Table 1.3 stands for the localization coordinates `X` and `Y` for the supplier and the customers.

	X	Y
Supplier	154	417
Customer 1	172	334
Customer 2	267	87
Customer 3	148	433
Customer 4	355	444
Customer 5	38	152

Table 1.3: `X` and `Y` localization coordinates

By the application of Theorem 1.3.1 and to be in accordance with the literature works, the distances are rounded to the nearest integer value to the sake of simplicity in the calculation. The Euclidian distance is then calculated by the Equation 1.19.

$$dist(i,j) = \left\lfloor \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \right\rceil \tag{1.19}$$

The details on the calculation of the distances from the customer ($i = 0$) are given below.

$$dist(0,0) = 0 \tag{1.20}$$
$$dist(0,1) = \left\lfloor \sqrt{(154-172)^2 + (417-334)^2} \right\rceil = \lfloor 84.93 \rceil = 85 \tag{1.21}$$
$$dist(0,2) = \left\lfloor \sqrt{(154-267)^2 + (417-87)^2} \right\rceil = \lfloor 348.81 \rceil = 349 \tag{1.22}$$
$$dist(0,3) = \left\lfloor \sqrt{(154-148)^2 + (417-433)^2} \right\rceil = \lfloor 17.09 \rceil = 17 \tag{1.23}$$
$$dist(0,4) = \left\lfloor \sqrt{(154-355)^2 + (417-444)^2} \right\rceil = \lfloor 202.81 \rceil = 203 \tag{1.24}$$
$$dist(0,5) = \left\lfloor \sqrt{(154-38)^2 + (417-152)^2} \right\rceil = \lfloor 289.28 \rceil = 289 \tag{1.25}$$

Note that for any pair of two customers (including or not the supplier) i and j, the distance $dist(i,j)$ is equal to the distance $dist(j,i)$, which allows to classify the matrix as symmetric. For this reason, given the distance $dist(i,j)$ for any given i and j, the distance $dist(j,i)$ calculation is not detailed above to avoid redundancy. The final matrix is given by Table 1.4.

	0	1	2	3	4	5
0	0,00	85,00	349,00	17,00	203,00	289,00
1	85,00	0,00	265,00	102,00	214,00	226,00
2	349,00	265,00	0,00	366,00	368,00	238,00
3	17,00	102,00	366,00	0,00	207,00	302,00
4	203,00	214,00	368,00	207,00	0,00	431,00
5	289,00	226,00	238,00	302,00	431,00	0,00

Table 1.4: Distance matrix

1.3.2 Routes scheduled

Once the distances among the customers and the supplier are established, the routes can be defined. A feasible route starts and ends at the supplier and is characterized by a sequence of customers for which a delivery operation is scheduled and at least one unit of product must be delivered. A vehicle is assigned to perform the trip and the total amount of products must not surpass its capacity.

Considering a set of customers $\{i, j, k, l, m, n, o, p\}$, a supplier (0) and the quantities $q = \{q_i, q_j, 0, 0, 0, q_n, 0, q_p\}$ requested to be delivered by the customers, a feasible route can be expressed by $\{0 \rightarrow j \rightarrow n \rightarrow p \rightarrow i \rightarrow 0\}$ which contains four from the eight available customers since those are the only for which a non zero quantity must be delivered. In order to obtain the cost of this route, the distance matrix is used and the calculation is given by $dist(0, j) + dist(j, n) + dist(n, p) + dist(p, i) + dist(i, 0)$.

From the guiding thread example, the transportation capacity per period is equal to 289. It means that for a single vehicle available per period, its capacity is equal to 289 but if there are two vehicles available, the capacity of each is of 144 rounding down to the nearest integer and so on (further explanation is given in Chapter 2, Section 2.4).

Thus, the optimal routes are the following:

Period 1 (1 vehicle needed)

Route 1

Sequence: $0 \rightarrow 1(65) \rightarrow 0$
Vehicle load: 65
Cost: $85 + 85 = 170$

Period 2 (2 vehicles needed)

Route 1

Sequence: $0 \rightarrow 3(116) \rightarrow 0$
Vehicle load: 116
Cost: $17 + 17 = 34$

Route 2

Sequence: $0 \rightarrow 4(48) \rightarrow 2(35) \rightarrow 5(22) \rightarrow 0$
Vehicle load: 105
Cost: $203 + 368 + 238 + 289 = 1098$

Period 3 (no vehicles needed)

No routes scheduled

Note that from the routing part and considering only the distance traveled and not the quantities in parenthesis, the transportation cost rises to 1302.

1.3.3 Deliveries and inventory levels

Considering the previous defined routes and the quantities to be delivered in parenthesis, the supplier and customers inventory levels can be calculated. Below, each table represents the evolution of the inventory level considering the deliveries and the associated costs.

Supplier						
t	s	r^t	$\sum q_0^t$	I_0^t	h_0^t	*cost*
0	510	-	-	510	0.30	153
1	510	193	65	638	0.30	191.4
2	638	193	221	610	0.30	183
3	610	193	0	803	0.30	240.9
					Total cost	768.3

Table 1.5: Supplier inventory calculation

Customer 1						
t	s	d_1^t	q_1^t	I_1^t	h_1^t	*cost*
0	130	-	-	130	0.23	29.9
1	130	65	65	130	0.23	29.9
2	130	65	0	65	0.23	14.95
3	65	65	0	0	0.23	0
					Total cost	74.75

Table 1.6: Customer 1 inventory calculation

Customer 2						
t	s	d_2^t	q_2^t	I_2^t	h_2^t	*cost*
0	70	-	-	70	0.32	22.4
1	70	35	0	35	0.32	11.2
2	35	35	35	35	0.32	11.2
3	35	35	0	0	0.32	0
					Total cost	44.8

Table 1.7: Customer 2 inventory calculation

Customer 3						
t	s	d_3^t	q_3^t	I_3^t	h_3^t	*cost*
0	58	-	-	58	0.33	19.14
1	58	58	0	0	0.33	0
2	0	58	116	58	0.33	19.14
3	58	58	0	0	0.33	0
					Total cost	38.28

Table 1.8: Customer 3 inventory calculation

Customer 4						
t	s	d_4^t	q_4^t	I_4^t	h_4^t	*cost*
0	48	-	-	48	0.23	11.04
1	48	24	0	24	0.23	5.52
2	24	24	48	48	0.23	11.04
3	48	24	0	24	0.23	5.52
					Total cost	33.12

Table 1.9: Customer 4 inventory calculation

Customer 5						
t	s	d_5^t	q_5^t	I_5^t	h_5^t	*cost*
0	11	-	-	11	0.18	1.98
1	11	11	0	0	0.18	0
2	0	11	22	11	0.18	1.98
3	11	11	0	0	0.18	0
					Total cost	3.96

Table 1.10: Customer 5 inventory calculation

The first table corresponds to the supplier and the column t to the period, s to its initial inventory level at period t, r^t to the production capacity per period, $\sum q_i^t$ to the total amount of products delivered to the set of customers, I_i^t to the inventory level at the end of period t, h_i^t to the inventory holding cost and *cost* to the associated inventory cost each period.

The following tables stand for the customers and column t corresponds to the period, s to the initial inventory level at period t, d_i^t to the customer demand at period t, q_i^t to the amount of products received from the supplier, I_i^t to the inventory level at the end of period t, h_i^t to the inventory holding cost and *cost* to the total customer inventory cost per period calculated as $I_i^t \times h_i^t$ for the supplier and the customers. The total cost value is calculated by $\sum_{t=0}^{3}(I_i^t \times h_i^t)$ in which $t = 0$ corresponds to the fictitious initial period in which no delivery operations are performed and $t = 3$ to the size of the horizon of time considered in this example.

Note that from the inventory part of the problem and considering the amount of products to be delivered and the inventory level at the end of each period, the inventory cost of this solution is equal to 963.21.

1.3.4 Solution cost

The solution cost can be split into the routing and the inventory parts. The first stands for the cost of the distance traveled and the seconds by the storage costs at the supplier or the customer. From the example, considering the data presented, the solution cost is expressed below.

$$\begin{aligned}
&\text{Solution cost} = \text{inventory cost} + \text{routing cost}\\
&\text{Solution cost} = (768.3 + 74.75 + 44.8 + 38.28 + 33.12 + 3.96) + (170 + 34 + 1098)\\
&\text{Solution cost} = (963.21) + (1302)\\
&\text{Solution cost} = 2265.21
\end{aligned}$$

Costs trade-off

A trade-off exists when regarding the inventory holding costs of the supplier and the customers. When there is a difference on these values and the cost of storage at a customer is lower than the supplier cost, there is a high probability of delivering more units of the product than what is needed by the customer, what generates a waste once the replenishment is done.

In such a situation, delivering more products to the customers lead to a more interesting situation since the inventory cost at the customer is lower than the supplier inventory cost. In other words, delivering more than the customers requirements can save the total inventory cost.

We may notice that in the guiding thread example, customers 1, 4 and 5 have an inventory holding cost (0.23, 0.23 and 0.18, respectively) which is lower than the supplier (0.30). The same does not occur for customers 2 and 3 (0.32 and 0.33, respectively). Let's analyse each one:

- Customer 1 inventory is replenished in period 1 only. The quantity is equivalent to the remaining space in the inventory, which considers the subtraction of the previous inventory level (130) and the maximum inventory level allowed of 195, which leads to 65 units of products. This value matches with the quantity delivered. At the end of the horizon, no waste is observed and it may be due to the high routing cost involved (compared to the inventory one) to visit this customer at another period.

- Customer 4 receives a delivery in period 2 only. The maximum inventory level allowed is of 72 and the remaining space available is equal to 48 considering the previous inventory level of 24 unities. Then, the delivery quantity matches the value of 48 products. In the case of this customer, there is a residual of 24 units at the end of the replenishment horizon. It means that even after treating all the customers assigned to the route, there is a remaining space in the vehicle and for that reason, and also because the inventory holding cost is lower than the supplier, more products than what is really needed is delivered and thus this waste is observed.

- Customer 5 is also delivered in period 2 only. Noticing that the remaining space is equal to 22 considering the maximum inventory level of 22 and no previous products in the inventory at the beginning of the period, 22 unities are then delivered. In the case of this customer, no remaining stock is observed at the end of the third period.

Thus, for any customer i and period t, if the inventory holding cost $h_i^t < h_0^t$, there is a tendance to deliver to the customer more than the quantity required if and only if the vehicle assigned has enough space and the customer inventory level remains under its maximum authorized level. Otherwise, if $h_i^t > h_0^t$, then the opposite occurs since there is no advantage in terms of cost to remove products from the depot to give to the customers.

It is important to highlight this cost trade-off because when solving the problem exactly, i.e., with a linear programming solver and without any time execution limit, it is more simple to obtain an optimal solution that considers the case of delivering more than what is needed when it reduces the inventory costs. Consequently, the solution cost tends to be closer or equal to the optimal solution easier. Other approach methods such as heuristics and metaheuristics do not take this situation into account. It may be necessary to add a pre-processing or a post-processing phase to these non-exact methods to handle such a situation that can lead to an slightly improvement in the solution cost.

1.4 Literature review

1.4.1 Inventory Routing Problem

Since many variants of the classical Inventory Routing Problem (IRP) exist, an extensive range of methods has been proposed in the literature to address these different variants. These methods vary significantly based on the type of algorithm developed. For instance, among the algorithms identified, we can categorize them into three groups: metaheuristics, matheuristics and exact approaches. Metaheuristics, such as those developed by Aksen et al. (2014) and Haddad et al. (2018), are widely recognized for their ability to provide high-quality solutions within a reasonable computational time by exploring a vast solution space. On the other hand, the matheuristics, like the ones proposed by Hemmati et al. (2016) and Touzout et al. (2022), combine the strengths of linear/mathematical programming and heuristic methods to improve the efficiency and accuracy of the solutions. Finally, exact approaches, which include methods developed by Manousakis et al. (2021), Archetti et al. (2020), and Coelho and Laporte (2014), aim to find optimal solutions by exhaustively searching the solution space, although these methods demand an intensive computational effort.

A significant number of works of the literature focuses on IRP models that consider a homogeneous fleet of vehicles as it is for the first paper (Archetti et al. (2007)) that models the IRP (named Vendor-Managed IRP) that also proposes the set of instances largely used in these days. This focus may be explained by the complexity involved in solving problems with a heterogeneous fleet of vehicles. Coelho and Laporte (2013, 2014) were among the pioneers in modeling IRP scenarios that account for both homogeneous and heterogeneous fleets. They developed sophisticated Branch-and-Cut (B&C) algorithms capable of addressing both fleet configurations. Despite this advancement, their results were limited to instances involving a homogeneous fleet of vehicles, possibly due to the computational challenges posed by heterogeneous fleets.

Cheng et al. (2017) introduced an IRP variant that incorporates environmental considerations, such as fuel consumption and carbon dioxide emissions, alongside a heterogeneous fleet of vehicles. Their work stands out as it is the only study in the literature that evaluates the proposed method for instances involving a heterogeneous fleet. The findings of Cheng et al. (2017) demonstrated that utilizing a heterogeneous fleet can lead to a reduction in total costs compared to a homogeneous fleet. However, it is important to note that the instances used in their study, which were based on those developed by Coelho and Laporte (2013), include specific characteristics related to environmental aspects, which may influence the generalizability of their results.

The exploration of these various methods and their applications highlights the diversity and complexity inherent in solving the IRP. Each approach offers unique advantages and challenges, reflecting the multiple nature of the problem and the ongoing efforts within the research community to develop more effective and efficient solutions.

(i) Green

To deal with the environmental aspects related to the emissions of pollutants on the supply chain, some authors have addresses the named Green IRP that incorporates features related to this scenario. It includes the comprehension on how these aspects occur in terms of emission rate and their inclusion into the problem formulation as it was done by Cheng et al. (2017). They added elements related to the fuel consumption and CO_2 emissions and concluded that for companies with high inventory holding costs, it is hard to control these emissions since more routing operations are needed and, consequently, more emission of pollutants are observed.

Soysal et al. (2018) proposed a mathematical formulation for a multi-supplier and perishable multi-product IRP with horizontal collaboration including the CO_2 emissions. They have applied the model to a 2-supplier 2-products case in order to show that when the suppliers collaborate with each other, the involved costs related to the problem and the emissions are reduced. Another multi-product approach was presented by Mirzapour Al-E-Hashem and Rekik (2014) and also considers transshipment. In Alinaghian et al. (2021), authors have added time windows to the problem and proposed multiple metaheuristics to solve it, including a Tabu Search (Glover 1990).

(ii) Perishable products

In the supply chain, some products that need to be transported are perishable and, consequently, can not stay too long on the shelter because they can perish and become useless. Rohmer et al. (2019) have shown a linear deterioration for the products and modeled it as a MILP and solved with a metaheuristic for a 2-echelon IRP. For the classical IRP, Alvarez et al. (2020) have considered a fixed shelf time for the products and introduced four mathematical formulations to solve the problem.

A survey on multiple papers that addresses the Perishable IRP (PIRP) was presented by Shaabani (2022). In this paper, the authors have classified the perishable products according to multiple characteristics, such as strict fixed lifetime, non-strict fixed lifetime, random shelf life and gradual deterioration by time, and have highlighted the fact that there are not many papers that treat this problem in the literature even if it is considered close to what happens in real scenarios.

(iii) Transshipment

Differently from the classical IRP in which a customer can be replenished only by the supplier, the IRP with transshipment (IRPT) authorizes the customers to perform delivery operations to another customer if it is beneficial for the solution cost. In practical, the linear models that deal with the transshipment operations need to consider an extra variable that represents the amount of products that is delivered directly from a supplier to a customer or from a customer to another one. Consequently, the inventory levels calculation depends on the classical variables that represent a normal delivery operation and a second one that considers the total transshipped. The IRPT has been first introduced by the works of Coelho et al. (2012b) and later, Lefever et al. (2018) have presented and strengthened a Branch-and-Cut formulation to improve the resolution quality of the problem, both for the single-product multi-vehicle case.

Azadeh et al. (2017) have introduced the perishable products category for the IRPT by considering a deterioration rate once the product is kept in the supplier of customer inventory. To solve the problem, a genetic algorithm is used. Later, the multi-product multi-vehicle case has been presented by Peres et al. (2017) with a real case study on a large Brazilian industry. Due to the problem complexity, a Randomized Variable Neighborhood Descent (RNVD) is used to solve the model.

(iv) Split delivery

This variant considers the fact that a customer can be visited more than once at each period. In this case, multiple deliveries can take place within the same period by different vehicles. In the non split version, at most one vehicle visits each customer at each period.

Applying the split delivery policy can be computationally hard but the results obtained by Dror and Trudeau (1990) for the Vehicle Routing Problem show that it can reduce the costs when the customers demands exceeds the vehicles capacity, which logically would lead to an unfeasible solution in the non split version of the problem. Also, Dinh et al. (2023) have added the split delivery constraints for the classical IRP version and also have shown a cost-saving scenario for both Maximum Level and Order-Up-to-Level inventory policies.

Yu et al. (2012) have tested the split delivery for the Stochastic IRP with service level constraints related to both supplier and customers and their hybrid solving approach based on Lagrangian relaxation is able to find high quality solutions in a reasonable time. Also, Li et al. (2011) have provided a study on the coordination of split deliveries in distribution systems which are similar to the IRP and have shown the importance of coordinating the split delivery with the inventory policy to avoid sub optimal solutions.

(v) Two-echelon

The two-echelon IRP (2E-IRP) consists of an IRP with an intermediate echelon that is responsible for its own delivery operations. The first echelon consists of the delivery of products from the supplier and the industry plants and the second by the plants and the customers. An example of a real-life application of the 2E-IRP was presented by Schenekemberg et al. (2020) for a petrochemical industry. In this case, authors consider the problem with fleet management (2E-IRPFM) that includes the decisions on how many vehicles to rent and when to provide the vehicle cleaning since the products transported are dangerous and present a risk of contamination.

As for the original one-echelon IRP, the 2E-IRP considers one supplier on its classical version. Authors Guimarães et al. (2019) have addressed the 2E-IRP with multiple suppliers (2E-MDIRP). In this case, the problem complexity increases since the number of delivery possibilities for the first echelon increases. They proposed a MILP and a branch-and-cut to solve the problem. Another solving approaches for the 2E-IRP considers branch-and-price algorithms as proposed by Charaf et al. (2024a), matheuristic as in Charaf et al. (2024b) and exact approaches considering multiple inventory policies as in Farias et al. (2019).

1.4.2 Split algorithms

Split-based methods have been widely used in the literature to solve various routing problems, showing their flexibility and effectiveness. Here, we reference several Split algorithms that have been proposed for the classical Vehicle Routing Problem (VRP) and those adapted for different VRP variants. Understanding how these methods work and why they produce high-quality solutions helps justify their use in solving the Heterogeneous Inventory Routing Problem with Batch Size (HIRP-BS).

In optimization problems, finding an initial solution and then improving it by exploring the local search space can be quite challenging. The Split algorithm, which inspired our method for solving the HIRP-BS, works in two main steps. The first step creates a sequence of customers, called a giant tour, that includes all customers without considering route limitations like vehicle capacities. In the second step, the giant tour is divided into several routes through the split phase, which solves a specific shortest path problem, and each route is assigned to a vehicle. The original idea of this approach was introduced by Beasley (1983) as a route-first, cluster-second method to solve the VRP. This method has became a key approach in solving routing problems, leading to many adaptations and improvements.

Since then, this approach has been widely used and developed in the literature. For example, Prins et al. (2014) provided a detailed list of around 70 papers that use this method in both basic and improved versions. The Split algorithm is very adaptable and can be classified into several categories based on its implementation, including:

(i) Basic versions of the method that follow the original principles of the Split algorithm

(ii) Simple extensions where some paths are excluded from the giant tour due to additional constraints from the specific optimization problem

(iii) Extensions considering shared resources, like vehicles with limited capacities or depots, which make the shortest path problem more complex

(iv) Special cases where the graphs have unique characteristics that require customized solutions.

These Split approaches can also be coupled with local search methods to improve solutions with efficient operators. This includes reconstructing feasible solutions into a giant tour and then using a perturbation step to avoid getting stuck in local minima and ensure diverse solutions. The process involves iterating between creating giant tours and refining feasible solutions, making it a flexible and dynamic problem-solving method.

Figure 1.15 shows an iterative schema of the Split algorithm presented by Prins et al. (2014). Starting from a giant tour, the sequence is split into a feasible solution, which can be further improved through local search moves. Once an improved solution is found, a regeneration phase, called Split^{-1}, creates a new giant tour that can be diversified using a perturbation operator. This results in another giant tour ready for further refinement. This procedure can be repeated multiple times, creating a strong iterative process that can be integrated into a metaheuristic framework to systematically enhance solution quality. The iterative nature of this process ensures that the algorithm continually adapts and improves, providing high-quality solutions to complex routing problems.

The wide use and adaptation of Split-based methods highlight their importance and effectiveness in solving routing problems. The ability to generate high-quality solutions through a structured yet flexible approach makes Split algorithms a valuable tool in the field of optimization and routing.

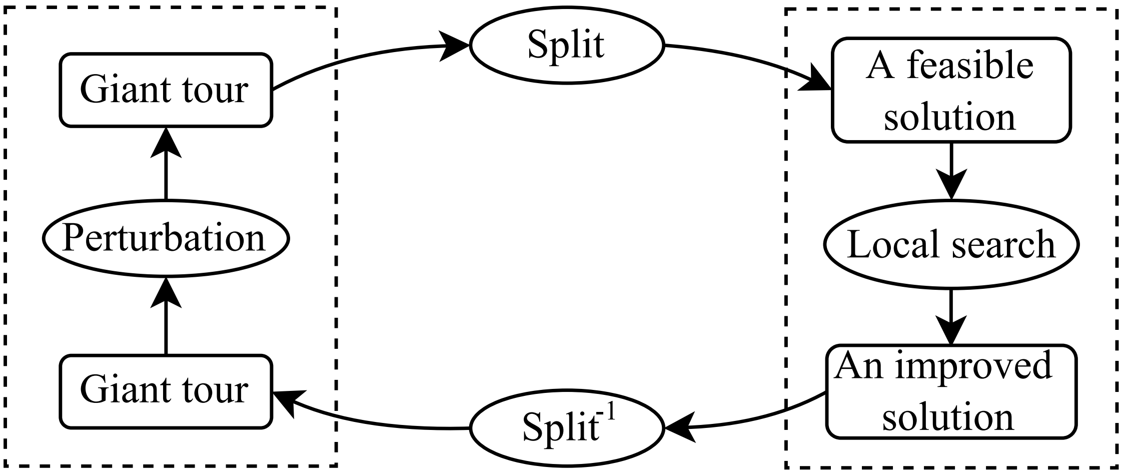


Figure 1.15: Alternation between search spaces (adapted from Prins et al. 2014)

Originally introduced by Dantzig and Ramser (1959) to address a routing and delivery problem for gasoline stations, the VRP has since become a widely studied problem due to its complexity and numerous variants, each applicable to different scenarios. The most common variant is the Capacitated VRP (CVRP), where a fleet of vehicles is tasked with meeting the demands of a set of customers. Prins (2004) developed a Split algorithm for solving the distance-constrained CVRP with an unlimited fleet of homogeneous vehicles. Their Split phase involves solving a shortest path problem on a directed acyclic graph derived from the giant tour. The minimum-cost solution for this problem is computed using the Bellman–Ford algorithm.

Algorithm 1 presents the Split for the CVRP and it was introduced by Prins et al. (2014). The notation include the giant tour T in which T_i corresponds to the customer assigned to the i^{th} position of T, V to the distance at each node of T, P to the parent node to help finding the critical path at the end, $c_{i,j}$ to the distance from i to j, s_i to the amount of products that must be delivered at i and Q to the vehicle capacity. Also, *load* and *cost* correspond to the vehicle load and the solution cost, respectively.

In Algorithm 1, in the beginning, the partial distances at each node of the giant tour are set to infinity (line 5). In the For loop, the next customer is retrieved (line 8) and the vehicle load is set to zero (line 9) since the current route does not contain any customer for the moment. In the Repeat loop (lines 10 to 23), the algorithm iterates until the end

Algorithm 1: Split for the CVRP

Result: V_j, P_j $\forall j \in \{1, \ldots, n\}$
1 **Begin**
2 $V_0 \leftarrow 0$
3 $P_0 \leftarrow 0$
4 **for** $i = 1$ ***to*** n **do**
5     $V_i \leftarrow \infty$
6 **end**
7 **for** $i = 1$ ***to*** n **do**
8     $j \leftarrow i$
9   load $\leftarrow 0$
10   **repeat**
11   load $\leftarrow$ load $+q_{T_j}$
12   **if** $i = j$ **then**
13   cost $\leftarrow c_{0,T_i} + s_{T_i} + c_{T_i,0}$
14   **end**
15   **else**
16   cost $\leftarrow$ cost $-c_{T_{j-1},0} + c_{T_{j-1},T_j} + s_{T_j} + c_{T_j,0}$
17   **end**
18   **if** *(load* $\leq Q$*)* ***and*** *(*$V_{i-1} + cost < V_j$*)* **then**
19             $V_j \leftarrow V_{i-1} + cost$
20             $P_j \leftarrow i - 1$
21   **end**
22         $j \leftarrow j + 1$
23   **until** $j > n$ ***or*** *load* $> Q$;
24 **end**
25 **End**

of the sequence is found ($j > n$) or if the current vehicle does not have enough capacity (*load* $> Q$). In this loop, the vehicle load is updated with the charge of T_j (line 11). Then, two cases can occur: if the customer is the first on the route (i.e., $i = j$), then the solution *cost* is updated with the cost of the arcs from the depot to this customer and from this customer to the depot as well as its amount of product are added (line 13); if not, which means that the current route has at least one customer, then the solution cost is updated by removing the cost of the arc linking the last customer and the depot to consider the new one as well as the amount of product to be delivered (line 16). If the vehicle load so far respect the vehicle capacity and adding the new customer can improve the best partial distance found so far, then the distance is improved (line 19) and the parent node is updated (line 20) considering the new customer T_j. Lastly, the customer index j is updated to the next one in the sequence of T (line 22).

Note that the number of vehicles is known only at the end of the algorithm and is equal to the number of routes created during the algorithm execution. It consists on the shortest path retrieval from the end to the beginning of the giant tour considering the parent nodes in P. Then, the solution cost correspond to the value V_n that consists of the distance calculated at the last node of the giant tour.

The Split algorithm can always generate a feasible solution for the CVRP with an unlimited fleet of homogeneous vehicles, provided that the highest customer demand is less than the highest vehicle capacity. In cases involving limited resources, the method becomes a shortest-path algorithm with resource consumption. Each node in the graph maintains a list of labels, where each label includes the solution cost and the remaining resources (e.g., vehicle availability and residual capacity at each node). The complexity of the Split phase increases with the number of candidate optimal solutions, as the number of labels generated at each node depends on the number and types of available resources. Consequently, depending on the giant tour and resource assignment throughout the Split phase, the algorithm may yield infeasible solutions if one or more required resources are unavailable for splitting the sequence entirely. Additionally, if multiple criteria or resources are considered at the end of the Split, it results in a Pareto front.

Duhamel et al. (2012) and Prins et al. (2014) adapted the original Split algorithm to the CVRP with a heterogeneous fleet of vehicles. Despite its limitations, their methods demonstrated promising results, which inspired us to employ a similar approach for solving the HIRP-BS.

□

A non-exhaustive selection of papers that have addresses the IRP and introduced new variants and approaches to solve it was made to show how new constraints can be incorporated to tackle different scenarios that relies on the the logistics characteristics. As already mentioned, the purpose of this thesis is to add new features that can turn the problem closer to a real scenario, including a heterogeneous vehicle fleet, inventory holding costs and customer demands that vary according to the each period and the delivery of products by batches rather than single units. These characteristics are relevant to be studied since some of them have never been or have barely been studied before.

The next two tables present some highlights on these selected papers by two different ways. Both are organized by alphabetical order of the first authors last name.

- Table 1.11 highlights in some words each paper contribution including the variant treated and the proposed approach to solve it and
- Table 1.12 presents the same papers but regarding the IRP version studied (classical or variant), the resolution method employed (heuristic, metaheuristic, hybrid or exact), the composition of the vehicle fleet (homogeneous or heterogeneous) and the variation on the inventory holding costs (period-independent or dependent).

Table 1.11: Bibliography synthesis on the IRP papers contributions

Citation	**Contribution**
Archetti et al. (2007)	Propose a branch-and-cut algorithm to solve the IRP (named Vendor-managed IRP) and introduces the benchmarking instances set to treat the single-vehicle homogeneous vehicle fleet case.
Coelho et al. (2012a)	Solve the IRP with a homogeneous vehicle fleet considering the multi-vehicle case up to five by adapting the classical IRP benchmarking instances initially designed for the single-vehicle case (Archetti et al., 2007).
Skålnes et al. (2023)	Introduce a new set of homogeneous instances for the IRP and solves them with a matheuristic and a branch-and-cut approaches.
Alvarez et al. (2020)	Four mathematical formulations, a branch-and-cut and a hybrid heuristic to the IRP with perishable products.
Aksen et al. (2014)	An IRP variant named Selective and Periodic IRP that considers collecting oil from a facility and aims to define the optimal periodic visits to be repeated as a cycle. An Adaptive Large Neighborhood Search algorithm is proposed to solve the problem.
Archetti et al. (2012)	A hybrid algorithm that combines a Tabu Search metaheuristic and a Mathematical Formulation to solve the classical IRP for the multi-vehicle homogeneous case per period.
Archetti et al. (2018)	Propose a branch-and-cut algorithm to solve the IRP with pickups and deliveries with a single homogeneous vehicle per period.
Archetti et al. (2020)	Propose a branch-and-cut to solve the IRP with pickups and deliveries for the multiple homogeneous vehicles per period.
Azadeh et al. (2017)	Introduce a genetic algorithm to solve the IRP considering perishable products and considers a single homogeneous vehicle per period.
Bertazzi et al. (2019)	Propose a matheuristic to solve an IRP variant with multiple depots and tests the algorithm with the classical IRP instances as well a new set introduced by them.

Continued on next page

Table 1.11: Table 1.11 – Bibliography synthesis on the IRP papers contributions (continued)

Citation	**Contribution**
Charaf et al. (2024b)	Introduces a two-phase matheuristic that combines the tabu search metaheuristic and a linear model to solve the classical 2E-IRP.
Charaf et al. (2024a)	Proposes a mathematical formulation and a branch-and-price to solve the classical version of the 2E-IRP.
Cheng et al. (2017)	Include environmental aspects to the classical IRP considering fuel and emission costs and considers driver wage. Proposes a mathematical formulation to take these aspects into account.
Coelho et al. (2012a)	Addresses the IRP considering consistency features that may help to take to problem closer to reality. These features include the vehicle capacity usage and intervals of delivery operations, for example. The problem is modelled as a MILP and a matheuristic is proposed to solve it.
Coelho et al. (2012b)	Introduce the IRP with transshipment which allows the deliveries to be done from the supplier or from the customer to another customer. Gives the related mathematical formulation and proposes a metaheuristic de type ALNS to solve it using the classical IRP instances.
Coelho et al. (2013)	Provide an extensive literature review on the IRP and its thirty years of new variants and methods to solve.
Coelho and Laporte (2013)	Propose an unified Branch-and-Cut algorithm to address several IRP variants and tests it for the classical IRP instances.
Coelho and Laporte (2014)	Introduce new valid inequalities to the IRP and some extensions and solve it using a Branch-and-Cut algorithm on the IRP classical instances.
Desaulniers et al. (2016)	Give a mathematical formulation and a Branch-and-Price-and-Cut to solve the classical IRP and improve the bounds on the classical instances set.
Diabat et al. (2024)	Propose a mathematical formulation and valid inequalities to solve the IRP with the Zero-Inventory-Ordering (ZIO) policy to analyse the impact of this inventory policy.
Dinh et al. (2023)	Considers the classical IRP version and incorporates split delivery constraints on both ML and OU inventory policies and show that the split delivery can save the inventory costs for the solutions found.

Continued on next page

Table 1.11: Table 1.11 – Bibliography synthesis on the IRP papers contributions (continued)

Citation	**Contribution**
Farias et al. (2019)	Introduces two formulations for the two-echelon IRP and adapts the classical IRP instances to handle the problem characteristics.
Guemri et al. (2016)	Solve the Multi-product Multi-vehicle IRP (MMIRP) with a Grasp metaheuristic and tests the algorithm with the classical instances.
Guimarães et al. (2019)	Proposes a mathematical formulation and a matheuristic to solve the two-echelon IRP with multiple depots (2E-MDIRP). In this problem, more than one supplier is considered as a source of deliveries on the first echelon.
Gutierrez-Alcoba et al. (2023)	Propose a MILP-based heuristic to solve a green IRP named Stochastic IRP on Electric Roads that include electric vehicles to perform the deliveries.
Hemmati et al. (2016)	Solve a sea IRP in which the vehicles are heterogeneous ships through a two-phase hybrid matheuristic.
Lefever et al. (2018)	Solves the IRP with Transshipment and proposes improvements on the literature mathematical formulation by introducing new valid inequalities and eliminating unnecessary variables, for example.
Manousakis et al. (2021)	Propose a branch-and-cut formulation for the classical IRP and introduces a new two-commodity flow formulation. This paper is considered so far the best benchmarking for the classical IRP instances using an exact formulation.
Mirzapour Al-E-Hashem and Rekik (2014)	Solve a multi-product IRP with transshipment with a MILP model.
Peres et al. (2017)	Addresses a multi-product IRP with transshipment and present a mathematical formulation and a metaheuristic to solve it. It presents a case study in a Brazilian retail industry.
Rohmer et al. (2019)	Solve the two-echelon IRP with perishable products through a three variant hybrid algorithm.
Schenekemberg et al. (2020)	Solve a two-echelon IRP incorporating fleet management (2E-IRPFM) that consists of decisions on renting and cleaning vehicles based on a real case of an industry. A mathematical formulation and a matheuristic are considered to solve the problem.

Continued on next page

Table 1.11: Table 1.11 – Bibliography synthesis on the IRP papers contributions (continued)

Citation	**Contribution**
Skålnes et al. (2023)	Incorporates features such as period-dependent inventory holding costs and different number of vehicles, for example, to the classical IRP. To solve the problem, they used a matheuristic and a branch-and-cut algorithm.
Soysal et al. (2015)	Solve an IRP with perishable products, uncertain demands and evaluation of CO_2 emission and fuel consumption by a MILP model.
Soysal et al. (2018)	Incorporate horizontal collaboration among supplier and customers to a green IRP considering perishable products comprising Key Performance Indicators (KIPs) related to the green aspects.
Touzout et al. (2022)	Solve the Time-Dependent IRP in which the distance traveled considered a starting time and not the distance traveled and solve it through a matheuristic by adapting the classical IRP set of instances.
Yu et al. (2012)	Solve the Stochastic IRP by adding service level constraints related to both supplier and customers. Considers the split delivery of the quantities to be delivered and solve it by a hybrid algorithm based on the Lagrangian relaxation.

Table 1.12: Overview on the IRP versions and approaches

Reference	IRP version		Resolution method			Vehicle fleet		Inventory holding costs	
	Classical	Variant	Meta/heuristic	Hybrid	Exact	Homogeneous	Heterogeneous	Period-independent	Period-dependent
Archetti et al. (2007)	✓				✓	✓		✓	
Coelho et al. (2012a)	✓			✓	✓	✓		✓	
Skålnes et al. (2023)	✓				✓	✓			✓
Alvarez et al. (2020)		✓		✓	✓	✓		✓	
Aksen et al. (2014)		✓	✓			✓		✓	
Archetti et al. (2012)	✓			✓		✓		✓	
Archetti et al. (2018)		✓			✓	✓		✓	
Archetti et al. (2020)		✓			✓	✓		✓	
Azadeh et al. (2017)		✓	✓			✓		✓	
Bertazzi et al. (2019)		✓		✓		✓		✓	
Charaf et al. (2024b)		✓		✓		✓		✓	
Charaf et al. (2024a)		✓			✓	✓		✓	
Cheng et al. (2017)		✓			✓		✓	✓	
Coelho et al. (2012a)		✓		✓		✓		✓	
Coelho et al. (2012b)		✓	✓		✓	✓		✓	
Coelho and Laporte (2013)		✓			✓	✓		✓	
Coelho and Laporte (2014)	✓	✓			✓	✓		✓	
Desaulniers et al. (2016)	✓				✓	✓		✓	
Diabat et al. (2024)		✓			✓	✓		✓	
Dinh et al. (2023)	✓				✓	✓		✓	
Farias et al. (2019)		✓			✓	✓		✓	
Guemri et al. (2016)		✓	✓			✓		✓	
Guimarães et al. (2019)		✓		✓	✓	✓		✓	

Continued on next page

Table 1.12 – Overview on the IRP versions and approaches (continued)

Reference	IRP version		Resolution method			Vehicle fleet		Inventory holding costs	
	Classical	Variant	Meta/heuristic	Hybrid	Exact	Homogeneous	Heterogeneous	Period-independent	Period-dependent
Gutierrez-Alcoba et al. (2023)		✓	✓			✓		✓	
Hemmati et al. (2016)		✓	✓	✓			✓	✓	
Lefever et al. (2018)		✓			✓	✓		✓	
Manousakis et al. (2021)	✓				✓	✓		✓	
Mirzapour Al-E-Hashem and Rekik (2014)		✓			✓		✓	✓	
Peres et al. (2017)		✓	✓		✓	✓		✓	
Rohmer et al. (2019)		✓		✓		✓		✓	
Schenekemberg et al. (2020)		✓		✓	✓	✓		✓	
Skålnes et al. (2023)	✓			✓	✓	✓			✓
Soysal et al. (2015)		✓			✓	✓		✓	
Soysal et al. (2018)		✓			✓	✓		✓	
Touzout et al. (2022)		✓		✓		✓		✓	
Yu et al. (2012)		✓		✓		✓		✓	
This thesis (2024)	✓	✓	✓		✓	✓	✓	✓	✓

As shown in Table 1.12, when regarding the IRP version studied, many variants have appeared due to the non sufficient classical constraints to address different scenarios that can arise. The resolution methods are mainly exact but since it is difficult to solve in a reasonable time, meta/heuristics and hybrid algorithms are also used. Concerning the vehicle fleet, papers that have addresses a heterogeneous fleet are rare and even if the formulation proposed the vehicles are indexed, the instances used present homogeneous vehicles. The same occurs with he inventory holding costs that are in the majority of the papers period-independent. These last two characteristics (vehicle fleet and inventory costs) deserve to be more emphasized and have motivated the choice of the features to be studied in this thesis, along with others that will be introduced in the next Chapter.

1.5 Chapter conclusion

In this chapter, the classical IRP has been defined in terms of the associated inventory management and routing problems. For each, the possible variations regarding, for example, the different supplier and customer points of view, the literature inventory policies and the different vehicle fleet compositions are presented. Later, the different parts of a feasible solutions are detailed and the calculations to define the distance matrix, the routes and inventory as well as the solution costs are detailed.

Then, a bibliography review on the different variants of IRP problems show the existing variations of the literature (Green, perishable products, transshipment and split delivery) and a review on the Split algorithms that have been proposed so far to treat the vehicle routing problems and some variations. It ends with two tables synthesis to summarize the most relevant papers that have treated the IRP and variants and to highlight the importance and the motivation to study another variation in this thesis.

Chapter remainder

- *Problem definition*
- *Problem analyses from the inventory management and routing points of view*
- *Definition of a feasible IRP solution and the involved calculations*
- *Bibliography review on the classical IRP and its variants as well as the resolution methods*
- *Bibliography review on the Split algorithms*

Chapter 2

Mathematical formulations

Go to the Table of Contents ↺

Abstract

In this Chapter, the mathematical formulations for the classical Inventory Routing Problem (IRP) and the new variant named Heterogeneous Inventory Routing Problem with Batch Size (HIRP-BS) are presented. The former stands for the classical version as it has been studied in the literature since 2007. The latter corresponds to a new variant that is first introduced in this work and consists in adding features that can turn the classical IRP closer to reality. One feature is related to the delivery operations in batches previously defined by the customers according to their needs. Another one is related to the inventory cost and demands varying over the time periods and the last one to the heterogeneous vehicle fleet that is also period-dependent and is characterized by a finite capacity and fixed and variable costs according to the distance traveled. A new set of instances to tackle the HIRP-BS is also introduced in this Chapter as well as its generation steps and explanations on the features addressed. Computational results are performed for both IRP and HIRP-BS linear formulations in order to test their performances in a set of literature benchmarking instances for the IRP, and a new set introduced in this work to handle the HIRP-BS variant.

Chapter content

2.1 Chapter introduction . . . 59
2.2 MILP formulation for the IRP . . . 59
    2.2.1 Variables . . . 59
    2.2.2 Objective function . . . 60
    2.2.3 Constraints . . . 61
2.3 MILP formulation for the HIRP-BS . . . 63
    2.3.1 Variables . . . 64
    2.3.2 Objective function . . . 65
    2.3.3 Constraints . . . 66
2.4 Instances . . . 68
    2.4.1 IRP classical instances . . . 68
    2.4.2 New HIRP-BS instances . . . 69
2.5 Experiments and results . . . 73
    2.5.1 Literature instances . . . 73

2.5.2 New benchmark instances 74
2.6 Chapter conclusion 76

2.1 Chapter introduction

The Inventory Routing Problem (IRP) and the Heterogeneous Inventory Routing Problem with Batch Size (HIRP-BS) can be modeled as Mixed Integer Linear Problems (MILP) to handle the constraints of transportation and inventory management simultaneously. In this Chapter, the corresponding formal description and the formulations for the IRP and the HIRP-BS are presented in Sections 2.2 and 2.3, respectively.

Also, the set of instances used are presented in Section 2.4. For the IRP, literature instances are considered. For the HIRP-BS, a new set of instances is introduced in this work because the existing ones are not well suited to handle the problem characteristics translated into the constraints including the heterogeneous fleet of vehicles and the batch sizes per customer.

Lastly, computational experiments and results are presented and discussed in Section 2.5 when solving both MILP models with an optimization solver.

2.2 MILP formulation for the IRP

The classical IRP consists in defining when and how many products have to be delivered to a set of customers in such a way that there is no stock disruption and that the customers inventory levels and the supplier production capacity is respected. To do so, a fleet os homogeneous vehicles are considered at each period of time and the routes to be performed also need to be defined. Customers demands and supplier production capacity are said to be deterministic. The objective is to define a minimum transportation and inventory cost solution taking into account a set of constraints regarding the inventory management, quantities to be delivered, sub-tours avoidance and vehicles capacity.

Formally, the IRP is defined on a graph $G = (\mathcal{N}', \mathcal{A})$ in which $\mathcal{N}$ corresponds to the set of n customers $\mathcal{N} = \{1, ..., n\}$ and the node 0 standing for the supplier, with $\mathcal{N}' = \{0\} \cup \mathcal{N}$. Therefore, $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}', i \neq j\}$ is the set of arcs. A time horizon $\mathcal{T} = \{1, ..., H\}$ with H periods is considered and consequently $\mathcal{T}' = \{0\} \cup \mathcal{T}$. A homogeneous fleet of m vehicles is considered, where each presents a capacity B. The distance matrix $C = (c_{i,j})_{0 \leq i,j \leq |\mathcal{N}'|}$ includes the cost $c_{i,j}$ to travel from i to j and respect triangular inequalities.

An initial inventory level s_i, $\forall i \in \mathcal{N}'$, is known in advance for the customers and supplier at period 0. The inventory holding costs are given by h_i^t, with $i \in \mathcal{N}'$ and $t \in \mathcal{T}'$. Each customer has a period-independent demand d_i^t that is the same for each period and a maximum inventory level U_i allowed per period $t \in \mathcal{T}$.

The sets, variables and data are summarized in Table 2.1.

The corresponding formulation is presented below and was inspired on the formulation presented by Archetti et al. (2014).

2.2.1 Variables

To model the IRP, the following variables are considered to handle the customers visitation, the quantities to be delivered, the flow and the inventory levels, respectively.

Table 2.1: Sets, data and variables

Sets/Data	Description
$\mathcal{N}$	Set of customers
$\mathcal{N}'$	Set of customers and the supplier
$\mathcal{A}$	Set of arcs (i,j), where $i,j \in \mathcal{N}'$ with $i \neq j$
$\mathcal{T}$	Set of T discrete time periods from 1 to $\lvert\mathcal{T}\rvert$
$\mathcal{T}'$	Set of T discrete time periods from 0 to $\lvert\mathcal{T}\rvert$
m	Number of vehicles available
B	Capacity of the vehicles
$c_{i,j}$	Distance from $i \in \mathcal{N}'$ to $j \in \mathcal{N}'$, $i \neq j$
r^t	Supplier production at time period $t \in \mathcal{T}$
d_i^t	Demand of customer $i \in \mathcal{N}$ in period $t \in \mathcal{T}$
U_i	Upper inventory level limit for customer $i \in \mathcal{N}$
s_i	Initial inventory level of $i \in \mathcal{N}'$
h_i^t	Inventory holding cost of $i \in \mathcal{N}'$ at time period $t \in \mathcal{T}'$
Variables	**Description**
$x_{i,j}^t$	Binary variable equal to 1 if arc $(i,j) \in A$ is chosen at $t \in \mathcal{T}$, 0 otherwise
$a_{i,j}^t$	Freight flow passing through arc $(i,j) \in A$ at period $t \in \mathcal{T}$
q_i^t	Quantity delivered to customer $i \in \mathcal{N}$ at period $t \in \mathcal{T}$
I_i^t	Inventory level of $i \in \mathcal{N}'$ at time period $t \in \mathcal{T}'$

- $x_{i,j}^t$: binary variable that indicates whether the customer or supplier $j \in \mathcal{N}'$ is preceded by $i \in \mathcal{N}'$ (with $i \neq j$) in a route performed in period $t \in \mathcal{T}$ ($x_{i,j}^t = 1$) or not ($x_{i,j}^t = 0$)

- q_i^t: the amount of products delivered to a customer $i \in \mathcal{N}$ at period $t \in \mathcal{T}$

- $a_{i,j}^t$: freight flow passing from customer or supplier $i \in \mathcal{N}'$ to $j \in \mathcal{N}'$ (with $i \neq j$) at period $t \in \mathcal{T}$

- I_i^t: inventory level of a customer or supplier $i \in \mathcal{N}'$ at period $t \in \mathcal{T}$.

2.2.2 Objective function

The objective function for the IRP consists in minimize the sum of the inventory and routing costs. The first considers the amount of products that are stored at the customers and the supplier and the second the costs regarding the total distance traveled to serve all the customers over all the periods of time. Commonly, both costs have similar importance when solving the problem and their details are presented below.

Inventory costs

The inventory costs consider the the inventory level I_i^t at the supplier and the customers at each period $t \in \mathcal{T}'$ and the inventory holding costs h_i^t associated as shown in Equation

2.1. Note that for period 0, the inventory level is equal to their initial inventory level, i.e., $I_i^0 = s_i \ \forall i \in \mathcal{N}'$.

$$\sum_{i \in \mathcal{N}'} \sum_{t \in \mathcal{T}'} h_i^t I_i^t \tag{2.1}$$

Routing costs

The routing costs are associated with the distance traveled by each homogeneous vehicle in order to visit the customers as in Equation 2.2. In this way, c_{ij} represents the costs to go from customer i to j and the binary variable $x_{i,j}^t$ defines wether a customer j is visited ($x_{i,j}^t = 1$) or not ($x_{i,j}^t = 0$) in period t.

$$\sum_{(i,j) \in \mathcal{A}} \sum_{t \in \mathcal{T}} c_{i,j} x_{i,j}^t \tag{2.2}$$

□

The objective is then to minimize the sum of these costs with similar weights or importance (Equation 2.3).

$$\min \sum_{i \in \mathcal{N}'} \sum_{t \in \mathcal{T}'} h_i^t I_i^t + \sum_{(i,j) \in \mathcal{A}} \sum_{t \in \mathcal{T}} c_{i,j} x_{i,j}^t \tag{2.3}$$

2.2.3 Constraints

Inventory level constraints

Initially (at period 0), the supplier and customers inventory levels variables I_i^t, $\forall i \in \mathcal{N}'$ and $t \in \mathcal{T}'$, are set to their initial inventory level s_i according to Constraints (2.4).

$$I_i^0 = s_i \qquad \forall i \in \mathcal{N}' \tag{2.4}$$

Then, in Constraints (2.5), the supplier inventory level is calculated considering the inventory level at the previous period I_0^{t-1}, $\forall t \in \mathcal{T}$, added of its production capacity $r^t, \forall t \in \mathcal{T}$ and deducts the total amount delivered $\sum_{i \in \mathcal{N}} q_i^t$, $\forall t \in \mathcal{T}$ to the customers.

$$I_0^t = I_0^{t-1} + r^t - \sum_{i \in \mathcal{N}} q_i^t \qquad \forall t \in \mathcal{T} \tag{2.5}$$

In Constraints (2.6), inventory level for each customer $i \in \mathcal{N}$ and period $t \in \mathcal{T}$ considers its precedent inventory level I_i^{t-1} and the amount of products delivered by the supplier and the demands for the current period.

$$I_i^t = I_i^{t-1} + q_i^t - d_i^t \qquad \forall i \in \mathcal{N}, \forall t \in \mathcal{T} \tag{2.6}$$

Delivery quantity constraints

The quantity q_i^t, $\forall i \in \mathcal{N}, t \in \mathcal{T}$ to be delivered must respect the remaining space available at the customer which is the difference between its previous inventory level and its maximum storage capacity ($U_i - I_i^{t-1} \ \forall i \in \mathcal{N}, t \in \mathcal{T}$) as in Constraints (2.7).

$$q_i^t \leq U_i - I_i^{t-1} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.7}$$

Constraints (2.8) requires that the customer receiving products at a time period be visited on one of the routes at the same period.

$$q_i^t \leq \min\{B, U_i\} \sum_{\substack{j \in \mathcal{N}' \\ j \neq i}} x_{i,j}^t \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.8}$$

Degree constraints

Also referred to as flow constraints, these ensure that the entering and leaving flow at each node is equal (Constraints 2.9).

$$\sum_{(i,j) \in \mathcal{A}} x_{i,j}^t = \sum_{(j,i) \in \mathcal{A}} x_{j,i}^t \qquad \forall i \in \mathcal{N}', t \in \mathcal{T} \tag{2.9}$$

Constraints (2.10) impose that the number m of available vehicles is respected by regarding the first arc $x_{0,i}^t$ $\forall i \in \mathcal{N}, t \in \mathcal{T}$ in a given route.

$$\sum_{i \in \mathcal{N}} x_{0,i}^t \leq m \qquad \forall t \in \mathcal{T} \tag{2.10}$$

Constraints (2.11) ensure that each customer is visited at most once per period since split deliveries are not allowed.

$$\sum_{\substack{j \in \mathcal{N}' \\ j \neq i}} x_{i,j}^t \leq 1 \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.11}$$

Vehicles capacity and sub-tour elimination constraints

In order to avoid sub-tours, variables $a_{i,j}^t$, $\forall i, j \in \mathcal{A}, t \in \mathcal{T}$, are introduced and serve as an increasing counter to ensure that a route starts and ends at node 0, which represents the supplier. All the variables $a_{i,0}^t$, $\forall i \in \mathcal{N}, t \in \mathcal{T}$ returning to the supplier are set to 0 as in constraints (2.12).

$$a_{i,0}^t = 0 \qquad \forall (i,0) \in \mathcal{A}, t \in \mathcal{T} \tag{2.12}$$

Constraints (2.13) are flow conservation constraints of the vehicle load when arriving and leaving each customer at each time period.

$$\sum_{\substack{j \in \mathcal{N}' \\ j \neq i}} a_{j,i}^t - q_i^t = \sum_{\substack{j \in \mathcal{N}' \\ j \neq i}} a_{i,j}^t \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.13}$$

Then, the vehicle load along each route is bounded by the vehicle capacity B as expressed in constraints (2.14).

$$a_{i,j}^t \leq B x_{i,j}^t \qquad \forall (i,j) \in \mathcal{A}, t \in \mathcal{T} \tag{2.14}$$

Domain of variables

Lastly, the domain of variables are given by Constraints (2.15)-(2.18).

$$x^t_{i,j} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}, t \in \mathcal{T} \tag{2.15}$$

$$a^t_{i,j} \geq 0 \qquad \forall (i,j) \in \mathcal{A}, t \in \mathcal{T} \tag{2.16}$$

$$q^t_i \geq 0 \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.17}$$

$$I^t_i \geq 0 \qquad \forall i \in \mathcal{N}', t \in \mathcal{T}' \tag{2.18}$$

2.3 MILP formulation for the HIRP-BS

The HIRP-BS is a multi-period IRP involving a heterogeneous fleet of vehicles and predefined batch sizes per customer. The supplier must determine when and how many batches of products to deliver to a set of customers and how to combine these customers into routes, ensuring that all customer demands are met and no inventory disruptions occur. Customer demands and supplier production are assumed to be deterministic and known for the entire time horizon.

Additionally, vehicle capacities and customer maximum inventory levels must be respected. The available vehicles may vary over the time horizon, and each customer can be visited at most once per time period (i.e., no split deliveries are allowed). The objective is to find a minimum-cost solution, with the cost comprising inventory holding and transportation expenses. The transportation cost includes fixed costs per vehicle type and variable costs based on distance traveled and vehicle type.

In mathematical terms, the HIRP-BS is defined on a graph $G = (\mathcal{N}', \mathcal{A})$ and considers: a set $\mathcal{N} = 1, 2, \ldots, n$ of customers; a single supplier denoted by 0, resulting in the overall set of vertices $\mathcal{N}' = \mathcal{N} \cup 0$; a time horizon set $\mathcal{T} = 1, \ldots, H$ of discrete time periods, where $\mathcal{T}' = \mathcal{T} \cup 0$; a period-dependent vehicle superset $\mathcal{K}$ and sets $\mathcal{K}^t$ ($\mathcal{K} = \mathcal{K}^1 \cup \mathcal{K}^2 \cup \cdots \cup \mathcal{K}^{|\mathcal{T}|}$) for each period $t \in \mathcal{T}$, with $|\mathcal{K}^t|$ different vehicle types available and $m^{k,t}$ vehicles of type $k \in \mathcal{K}^t$. Each vehicle type $k \in \mathcal{K}^t$ has a capacity and associated fixed and variable costs given by $\mathcal{B}^{k,t}$, $f^{k,t}$, and $v^{k,t}$, respectively. For convenience, vehicle types are sorted by increasing capacity, i.e., $\mathcal{B}^{k,t} \leq \mathcal{B}^{k+1,t}$ for all $k \in \mathcal{K}^t, k \neq |\mathcal{K}^t|$ and for all $t \in \mathcal{T}$.

An arc $(i,j) \in \mathcal{A}$ is created for each pair of vertices $i \in \mathcal{N}'$ and $j \in \mathcal{N}'$, with $i \neq j$. The distance matrix is given by $C = (c_{i,j})0 \leq i,j \leq |\mathcal{N}'|$, and the cost associated with arc (i,j) is defined by $c_{i,j}$. Another matrix $S = (s_{i,j})0 \leq i,j \leq |\mathcal{N}'|$ represents the shortest path from i to j to handle non-Euclidean distances (when applicable). Note that this value may differ from the Euclidean distance. The variable cost for sending a vehicle from i to j is computed as $v^{k,t} \times c_{i,j}$.

Each customer $i \in \mathcal{N}$ has a demand d^t_i at period $t \in \mathcal{T}$ and a maximum inventory level U_i that is period-independent. The supplier produces r^t units of products per period t. An initial inventory level s_i, $i \in \mathcal{N}'$, is observed at each customer and at the supplier, and each customer has a predefined batch size ℓ_i that requires the quantities delivered to be multiples of this value. Additionally, period-dependent inventory holding costs are denoted as h^t_i for each vertex $i \in \mathcal{N}'$ at period $t \in \mathcal{T}$. The inventory costs are calculated by multiplying h^t_i by the inventory level for each $i \in \mathcal{N}'$ and $t \in \mathcal{T}$. We impose $h^0_i = h^1_i$ for any $i \in \mathcal{N}'$.

The HIRP-BS is formulated by adapting the flow formulation introduced by Archetti et al. (2014) for the classical IRP. To account for the specific characteristics of the HIRP-BS, new constraints are added to address the heterogeneous fleet of vehicles available per period and the customer batch sizes.

To model the HIRP-BS, the following variables are considered: a binary variable $x_{i,j}^{k,t}$ that is equal to 1 if $i \in \mathcal{N}'$ is followed by $j \in \mathcal{N}'$, $i \neq j$, in a route performed at period $t \in \mathcal{T}$ by a vehicle of type $k \in \mathcal{K}^t$, and 0 otherwise; a continuous variable $q_i^{k,t}$ that defines the quantity delivered to customer $i \in \mathcal{N}$ at period $t \in \mathcal{T}$ by a vehicle of type $k \in \mathcal{K}^t$; an integer variable z_i^t corresponding to the number of batches delivered to customer $i \in \mathcal{N}$ at period $t \in \mathcal{T}$ (such that $q_i^{k,t} = \ell_i z_i^t$); an integer variable I_i^t representing the inventory level of each vertex $i \in \mathcal{N}'$ at period $t \in \mathcal{T}'$; and an integer variable $a_{i,j}^{k,t}$ representing the total flow from $i \in \mathcal{N}'$ to $j \in \mathcal{N}'$, $i \neq j$, at period $t \in \mathcal{T}$ using a vehicle of type $k \in \mathcal{K}^t$.

Sets/Data	Description
$\mathcal{N}$	Set of customers
$\mathcal{N}'$	Set of customers and the supplier
$\mathcal{A}$	Set of arcs (i,j), where $i,j \in \mathcal{N}'$ with $i \neq j$
$\mathcal{T}$	Set of T discrete time periods from 1 to $\|\mathcal{T}\|$
$\mathcal{T}'$	Set of T discrete time periods from 0 to $\|\mathcal{T}\|$
$\mathcal{K}$	Unique indexes set of available vehicles from 1 to $\|\mathcal{T}\|$
$m^{k,t}$	Number of vehicles of type $k \in \mathcal{K}^t$ available at period $t \in \mathcal{T}$
$\mathcal{K}^t$	Set of vehicles available at time period $t \in \mathcal{T}$, where $\mathcal{K}^t \subseteq \mathcal{K}$ and $\mathcal{K}^t \cap \mathcal{K}^{t'} = \varnothing$, $\forall t' \in \mathcal{T}$, $t' \neq t$
$B^{k,t}$	Capacity of vehicle $k \in \mathcal{K}^t$ at period $t \in \mathcal{T}$
f^k	Fixed cost for vehicle $k \in \mathcal{K}^t$ of period $t \in \mathcal{T}$
v^k	Cost per unit of distance traveled by vehicle $k \in \mathcal{K}^t$ of period $t \in \mathcal{T}$
$c_{i,j}$	Parameter distance from $i \in \mathcal{N}'$ to $j \in \mathcal{N}'$, $i \neq j$
$s_{i,j}$	Calculated shortest path from $i \in \mathcal{N}'$ to $j \in \mathcal{N}'$, $i \neq j$
r^t	Supplier production at time period $t \in \mathcal{T}$
d_i^t	Demand of customer $i \in \mathcal{N}$ at time period $t \in \mathcal{T}$
ℓ_i	Batch size for customer $i \in \mathcal{N}$
U_i	Upper inventory level limits for customer $i \in \mathcal{N}$
s_i	Initial inventory level of $i \in \mathcal{N}'$
h_i^t	Inventory holding cost of $i \in \mathcal{N}'$ at time period $t \in \mathcal{T}'$
Variables	**Description**
$x_{i,j}^{k,t}$	Binary variable equal to 1 if arc $(i,j) \in \mathcal{A}$ is chosen at $t \in \mathcal{T}$ by $k \in \mathcal{K}^t$, 0 otherwise
$a_{i,j}^{k,t}$	Freight flow passing through arc $(i,j) \in \mathcal{A}$ at period $t \in \mathcal{T}$ in vehicle $k \in \mathcal{K}^t$
$q_i^{k,t}$	Quantity delivered to customer $i \in \mathcal{N}$ at period $t \in \mathcal{T}$ using vehicle $k \in \mathcal{K}^t$
z_i^t	Number of batches delivered to customer $i \in \mathcal{N}$ at time period $t \in \mathcal{T}$
I_i^t	Inventory level of $i \in \mathcal{N}'$ at time period $t \in \mathcal{T}'$

Table 2.2: Description of sets, data and variables

2.3.1 Variables

To model the HIRP-BS, the following variables are considered to handle the customers visitation, the quantities and the number of batches to be delivered, the flow and the inventory levels, respectively.

- $x_{i,j}^{k,t}$: binary variable that indicates whether the customer $j \in \mathcal{N}'$ is preceded by $i \in \mathcal{N}'$ (with $i \neq j$) in a route performed by a vehicle type $k \in \mathcal{K}^t$ in period $t \in \mathcal{T}$

($x_{i,j}^{k,t} = 1$) or not ($x_{i,j}^{k,t} = 0$);

- $q_i^{k,t}$: the amount of products delivered to a customer $i \in \mathcal{N}$ in a vehicle type $k \in \mathcal{K}^t$ available at period $t \in \mathcal{T}$;
- z_i^t: the number of batches delivered to a customer $i \in \mathcal{N}$ at period $t \in \mathcal{T}$;
- $a_{i,j}^{k,t}$: freight flow passing from customer $i \in \mathcal{N}'$ to $j \in \mathcal{N}'$ (with $i \neq j$) in a vehicle type $k \in \mathcal{K}^t$ at period $t \in \mathcal{T}$;
- I_i^t: inventory level of a customer $i \in \mathcal{N}'$ at period $t \in \mathcal{T}'$.

2.3.2 Objective function

Given that the objective of the HIRP-BS is to minimize the sum of the inventory and routing costs, the objective function can be split in two parts for the sake of comprehension. The former considers the supplier and customers inventory holding costs over all periods of time (including period 0) and the latter the routing costs related to the total distance traveled considering both fixed and variable costs according to the vehicle types used in a given route. As for the IRP, both costs have similar importance when solving the problem and their details are presented below.

Inventory costs

At first, inventory costs hold at each time period from 0 (to consider both customers and supplier initial inventory level, *i.e.*, $I_i^0 = s_i$) to $|\mathcal{T}|$ and contemplate the cost per unit h_i^t and the inventory level I_i^t according to Equation (2.19).

$$\sum_{i \in \mathcal{N}'} \sum_{t \in \mathcal{T}'} h_i^t I_i^t \tag{2.19}$$

Routing costs

Second, routing costs involve fixed and variable costs according to the use of a specific vehicle type and the distance traveled, respectively. Both are detailed below by Equations (2.20) and (2.21).

Fixed costs

In order to add the fixed costs ($f^{k,t}$) per vehicle type and period, it is necessary to know which type is assigned to a given route and which period. Also, to add the fixed cost to the objective function, we consider getting the first arc of a route, *i.e.*, the one that starts at the supplier ($i = 0$) and go to any customer $j \in \mathcal{N}$ at any period and using any vehicle type according to Equation (2.20).

$$\sum_{j \in \mathcal{N}} \sum_{t \in \mathcal{T}} f^{k,t} x_{0,j}^{k,t} \tag{2.20}$$

Variable costs

Since the variable costs ($v^{k,t}$) are considered based on each unit of distance traveled ($s_{i,j}$) per arc $(i, j) \in A$, we need to consider all the arcs (*i.e.*, when $x_{i,j}^{k,t} = 1$) at each route performed by a given vehicle type in a period of time according to Equation (2.21).

$$\sum_{(i,j)\in A}\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}^t} s_{i,j}v^{k,t}x_{i,j}^{k,t} \tag{2.21}$$

□

The objective is then to minimize the sum of these costs with similar weights or importance (Equation 2.22).

$$minimize \quad \sum_{i\in\mathcal{N}'}\sum_{t\in\mathcal{T}'} h_i^t I_i^t + \sum_{(i,j)\in A}\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}^t} s_{i,j}v^{k,t}x_{i,j}^{k,t} + \sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} f^{k,t}x_{0,j}^{k,t} \tag{2.22}$$

2.3.3 Constraints

The constraints can be classified in four groups according to their major relevance to the formulation: classical inventory management, delivery quantities, degree and capacity of vehicles and sub-tour elimination. Each of them is presented below by Equations (2.23)-(2.40).

Inventory level constraints

Classical inventory management constraints are given by Constraints (2.23) to (2.25).

In constraint (2.23), the inventory level at period zero (I_i^0) for each customer and the supplier is equal to their initial inventory level which is a parameter of the problem (s_i).

$$I_i^0 = s_i \qquad \forall i \in \mathcal{N}' \tag{2.23}$$

The supplier inventory level at a period t is given by its level in the precedent period $t-1$ added to its production capacity in t given by r^t minus the amount of products delivered to each customer visited in t according to Equations (2.24).

$$I_0^t = I_0^{t-1} + r^t - \sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{K}^t} q_i^{k,t} \qquad \forall t \in \mathcal{T} \tag{2.24}$$

Similarly, for the customers, their inventory level in period t is given by its level in $t-1$ added by the quantity received from the supplier minus the customer demand as expressed by Equations (2.25).

$$I_i^t = I_i^{t-1} + \sum_{k\in\mathcal{K}^t} q_i^{k,t} - d_i^t \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.25}$$

Delivery quantities constraints

The quantities delivered are modeled by Constraints (2.26) to (2.28).

In a given period, each customer can be delivered at most by a single vehicle from those available. In order to avoid stock disruption, this quantity must respect maximum (U_i) inventory level allowed per customer and the existing inventory level (I_i^{t-1}), which is imposed by constraints (2.26).

$$\sum_{k\in\mathcal{K}^t} q_i^{k,t} \leq U_i - I_i^{t-1} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.26}$$

The quantity of products delivered to a customer ($q_i^{k,t}$) is bounded by the minimum value between the vehicle capacity (B^k) and the customer maximum inventory level (U_i) according to the inequalities (2.27).

$$q_i^{k,t} \leq \min\{B^k, U_i\} \sum_{\substack{j \in \mathcal{N}' \\ j \neq i}} x_{i,j}^{k,t} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.27}$$

Then, to take into account the batch size ℓ_i per customer, we introduce constraints (2.28) to ensure that quantities delivered are a multiple of this value.

$$\sum_{k \in \mathcal{K}^t} q_i^{k,t} = z_i^t \ell_i \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.28}$$

Degree (flow) constraints

The degree constraints (also named flow constraints) are given by Equations (2.29) to (2.32).

Constraints (2.29) guarantee that the number of vehicles arriving and leaving the customers or supplier is the same.

$$\sum_{(i,j) \in A} x_{i,j}^{k,t} = \sum_{(j,i) \in A} x_{j,i}^{k,t} \qquad \forall i \in \mathcal{N}', t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.29}$$

Constraints (2.30) forbid the split delivering and impose that a customer is visited at most once per period.

$$\sum_{\substack{j \in \mathcal{N}' \\ j \neq i}} \sum_{k \in \mathcal{K}^t} x_{i,j}^{k,t} \leq 1 \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.30}$$

Constraints (2.31) are introduced to limit to one the number of routes per period performed by each available vehicle.

$$\sum_{i \in \mathcal{N}} x_{0,i}^{k,t} \leq 1 \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.31}$$

In order to impose that a customer is visited only if a non-zero quantity of products is delivered, we added constraints (2.32).

$$\sum_{\substack{j \in \mathcal{N}' \\ j \neq i}} \sum_{k \in \mathcal{K}^t} x_{i,j}^{k,t} \leq \sum_{k \in \mathcal{K}^t} q_i^{k,t} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.32}$$

Vehicles capacity and sub-tour elimination constraints

To avoid sub-tours in the solutions, constraints (2.33) to (2.35) are added and represent the Miller-Tucker-Zemlin (MTZ) (Miller et al., 1960) constraints for the multi vehicle routing problem associated.

Constraints (2.33) require that each vehicle at each time period returns empty (no flow) to the supplier.

$$a_{i,0}^{k,t} = 0 \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.33}$$

Constraints (2.34) are flow conservation constraints of the vehicle load when arriving and leaving each customer at each time period.

$$\sum_{\substack{j \in \mathcal{N}' \\ j \neq i}} a_{j,i}^{k,t} - q_i^{k,t} = \sum_{\substack{j \in \mathcal{N}' \\ j \neq i}} a_{i,j}^{k,t} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.34}$$

Constraints (2.35) ensure consistency between the values of the $a_{i,j}^{k,t}$ and $x_{i,j}^{k,t}$ variables and impose that the capacities of the vehicles are respected.

$$a_{i,j}^{k,t} \leq B^k x_{i,j}^{k,t} \qquad \forall (i,j) \in A, t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.35}$$

Variables domain

At last, variables domain are provided by constraints (2.36)–(2.40).

$$x_{i,j}^{k,t} \in \{0,1\} \qquad \forall (i,j) \in A, t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.36}$$

$$a_{i,j}^{k,t} \geq 0 \qquad \forall (i,j) \in A, t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.37}$$

$$q_i^{k,t} \geq 0 \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.38}$$

$$z_i^t \in \mathbb{Z}^+ \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.39}$$

$$I_i^t \geq 0 \qquad \forall i \in \mathcal{N}', t \in \mathcal{T}' \tag{2.40}$$

2.4 Instances

Two groups of the IRP data are distinguished in this work. The former accounts for the literature classical instances and the latter for the new set created in this thesis to deal with the HIRP-BS variant.

2.4.1 IRP classical instances

Literature instances were proposed by Archetti et al. (2007). These authors presented a total of 160 files varying from 5 to 50 customers and 3 to 6 periods of time each. These instances are characterized by a period independent demand and inventory costs over the periods as well as the Euclidian distance among the customers. An example of an instance file is given in Appendix A, Section A.1.

Note that for these classical instances, the demands and the inventory costs are the same over the whole time horizon and that the initial inventory level, the maximum storage capacity and the demands are multiple between them.

Table 2.3 gives an overview on these 160 instances grouping them by number of customers ($\mathcal{N}$), number of periods of time ($\mathcal{T}$), customers and supplier inventory holding costs, respectively (h_i, h_0).

Each file is named `absXnY` in which `X` corresponds to the type of instance and since each instance size has five different combinations, `X` values range from 1 to 5. `Y` values

Table 2.3: Literature instances characteristics

Number of instances			$\mathcal{N}$	$\|\mathcal{T}\|$	h_i	h_0
160	100	50	{5, 10, ..., 50}	3	[0.01; 0.05]	0.03
		50	{5, 10, ..., 50}	3	[0.1; 0.5]	0.3
	60	30	{5, 10, ..., 30}	6	[0.01; 0.05]	0.03
		30	{5, 10, ..., 30}	6	[0.1; 0.5]	0.3

corresponds to the number of customers, which means that for the 3-period instances set, the possible values are equal to $5c$, with $c = \{1, 2, ..., 10\}$ and for the 6-period ones, these values are up to $5c$, with $c = \{1, 2, ..., 6\}$.

Initially, they were designed to treat the single vehicle case and later, authors Archetti et al. (2012); Coelho et al. (2012a) have addressed the multi vehicle case up to 5 vehicles. For each instance, the capacity of the single vehicle is indicated. To tackle the multi vehicle case, this value is better interpreted as a transportation capacity per period and not per vehicle since the value is divided by the number of available vehicles and the result is rounded down to the nearest integer. Then, the number of instances increase to 800 since the 160 base instances can be now used for up to 5 vehicles each. However, there is only one exception in which the instance `abs1n5` becomes infeasible when using 5 vehicles since the vehicles transportation capacity is not enough to handle the customers demands over the periods.

2.4.2 New HIRP-BS instances

Several sets of instances for the classic IRP were previously presented in the literature. In their majority, a relatively short time horizon of up to 6 periods is considered, except for instances proposed by Skålnes et al. (2023) with a time horizon of 12 periods. The most commonly considered instances are those introduced by Archetti et al. (2007). Available instances usually consider constant demands inventory holding cost over the time horizon, except for instances presented in Skålnes et al. (2023), which contain period-dependent demands. Table 2.4 summarizes instance parameters for those present in the literature and the new set of instances proposed by us in the last column.

	Archetti et al. (2007)	Archetti et al. (2012)	Coelho et al. (2012b)	Skålnes et al. (2023)	**This thesis (2024)**
Number of customers	5–50	50–200	5–50	10–200	**19–183**
Number of periods	3,6	6	3,6	6, 9, 12	**7, 14, 21, 28**
Period-dependent inventory costs	No	No	No	No	**Yes**
Period-dependent demands	No	No	No	Yes	**Yes**
Heterogeneous fleet of vehicles	No	No	No	No	**Yes**
Batch size per customer	No	No	No	No	**Yes**
Euclidian distance	Yes	Yes	Yes	Yes	**No**

Table 2.4: Comparison of instance features

Since the instances introduced in these papers can not handle the HIRP-BS variant and are quite far from its characteristics, a new set of 80 instances is introduced in this thesis.

These instances consider an heterogeneous vehicle fleet, demands and inventory holding costs that vary over the time horizon and a batch size per customer. Their characteristics are detailed below.

1. Number of customers and periods: a number of customers of up to 183 and a time horizon of up to 28 periods
2. Heterogeneous vehicle fleet: period-dependent vehicle types are available at each period and each has a quantity, a capacity, fixed and variable costs depending on their usage and the distance traveled, respectively
3. Demands and inventory holding costs: period-dependent values for each period of time which represents possible fluctuations due to the customers constraints and different inventory costs according to, for example, the difficulty in their storage
4. Customers batch sizes: each customer has a period-independent fixed batch size which consists in a minimum products group size to be delivered at each period and to facilitate their management and/or storage
5. Non Euclidian distances: real-life distances between French cities in which the distances are non-Euclidean and triangle inequalities are not necessarily satisfied.

In order to generate these instances, the following steps were employed. Note that some of these instances may provide infeasibilities and in some cases, the solver detected them instantly at the beginning of the execution. In these cases, the instances concerned were discarded.

- Customer demand (d_i^t)

$$rand([10, 100]) \in \mathbb{Z}^+ \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.41}$$

- Inventory cost (h_i^t)

$$rand([0.1, 0.5]) \in \mathbb{R}^+ \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.42}$$

- Number of vehicle types per period ($|\mathcal{K}^t|$)

$$rand([1, 6]) \in \mathbb{Z}^+ \qquad \forall t \in \mathcal{T} \tag{2.43}$$

- Number of vehicles of each type ($m^{k,t}$)

$$rand([1, 5]) \in \mathbb{Z}^+ \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.44}$$

- Capacity of vehicle type k at period t ($B^{k,t}$)

$$\left\lceil \frac{\sum_{i \in N} d_i^t}{|\mathcal{K}^t|} \times (rand([1, 1.5]) \in \mathbb{R}^+) \right\rceil \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.45}$$

- Fixed cost for vehicle type k at period t ($f^{k,t}$)

$$rand([50, 150]) \in \mathbb{Z}^+ \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.46}$$

- Variable cost for vehicle type k at period t ($v^{k,t}$)

$$rand([0.5, 3]) \in \mathbb{R}^+ \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}^t \tag{2.47}$$

- Maximum inventory allowed for customer i (U_i)

$$\lceil max(d_i^t) \times (rand([1, 2]) \in \mathbb{R}^+) \rceil \qquad \forall i \in \mathcal{C}, t \in \mathcal{T} \tag{2.48}$$

- Initial inventory level for customer i (s_i)

$$\lceil U_i \times (rand([0.2, 1]) \in \mathbb{R}^+) \rceil \qquad \forall i \in \mathcal{N} \tag{2.49}$$

- Batch size for customer i (ℓ_i)

$$\left\lceil \frac{d_i^t}{(rand([5, 8]) \in \mathbb{Z}^+)} \right\rceil \qquad \forall i \in \mathcal{N}, t \in rand([1, |\mathcal{T}|]) \tag{2.50}$$

- Supplier production at period t (r^t)

$$\sum_{i \in \mathcal{N}} d_i^t \times (rand([1.2, 2]) \in \mathbb{R}^+) \qquad \forall t \in \mathcal{T} \tag{2.51}$$

- Inventory holding cost of supplier at period t (h_0^t)

$$rand([0.1, 0.5]) \in \mathbb{R}^+ \qquad \forall t \in \mathcal{T} \tag{2.52}$$

- Initial inventory level of supplier (s_0)

$$\frac{\sum_{i \in \mathcal{N}} s_i^0}{2} \tag{2.53}$$

The 80 selected instances are all feasible, which means that at least one solution exists to meet all the problem constraints. These instances can then be divided into groups according to their size in terms of number of customers and of periods of time. Three groups are then created: Small, Medium and Large. Table 2.5 presents the classification.

Table 2.5: Sets of instances for the HIRP-BS

	Small-scale	Medium-scale	Large-scale
Number of instances	13	57	10
Number of periods	7	7, 14, 21, 28	7, 14, 21, 28
Number of customers	19	19*, 34, 46, 58, 83	114, 149, 170, 183

(*) Except for a time horizon with 7 periods

These new instances are up to 183 customers and 28 periods of time, which are challenging when compared to those that exists in the literature. They were grouped according to their facility of resolution and are presented below.

- Small set: this set contains instances with 19 customers and a time horizon of 7 periods. Instances from this set are the only ones for which feasible solutions can be found within a given running time limit by the proposed formulation using an MILP solver (further presented in this Chapter)
- Medium set: this set contains instances with 19, 34, 46, 58, and 83 customers and a time horizon with 7 (except for a number of customers equal to 19), 14, 21, and 28 time periods. These instances do not have an integer feasible solution known by a MILP solver but only for the metaheuristic approach (further presented in the Chapter 4). Their performance is then established between the linear relaxation value provided by a MILP solver and the integer feasible solution values from the metaheuristic.
- Large set: these instances provide 114, 149, 170 and 183 customers and 7, 14, 21 and 28 periods of time. They were grouped together since as for the medium set, an integer feasible solution is not known but only the linear relaxation and in terms of the heuristic approach, only a part of the metaheuristic can be employed since its dimensions make their resolution complicated for the whole method.

These instances can be downloaded at `https://perso.limos.fr/~diperdigao/research/HIRP-BS/instances/`. A file example is presented in Appendix A, Section A.2.

The distances matrices $C = (c_{i,j})_{0 \leq i,j \leq |\mathcal{N}'|}$ were extracted from the set of instances proposed by Duhamel et al. (2012) for the HVRP and were extracted with an API that calculates the distance according to the time required to travel from one point to another. These distances may not be Euclidian in some cases and, consequently, the triangle inequalities are not verified.

□

Theorem 2.4.1 (Triangle inequalities) *For any triplet of points A, B, C such that $AB \leq AC + CB$, the triangle inequalities state that*

$$AB + BC > CA$$
$$BC + CA > AB$$
$$BA + AC > CB$$

Once one of these inequalities is not verified, we can state that the distances are not Euclidian since the Euclidian distance corresponds to the shortest path between two points, i.e., always verify the three inequalities.

□

One example from these real distance matrices can be taken from the medium instance $m\_19\_14\_1$ with 19 customers and 14 periods of time. Taking a look at the distances between customers 1, 3 and 8, we can see that traveling from 3 to 8 is faster if we pass at customer 1 location as shown in Figure 2.1.

To tackle the cases in which these inequalities are not verified, a pre-processing before using them in order to find the shortest path among the customers and the supplier and can be easily achieved with the Dijkstra (Dijkstra, 1959) shortest path algorithm. It consists in executing the algorithm using the original distance matrix and updating it once a shortest path is found compared to the actual values.

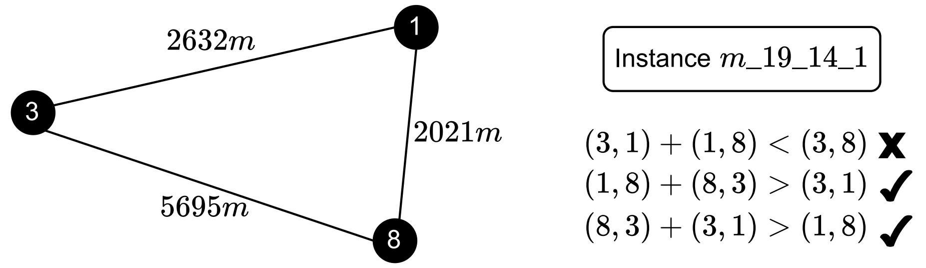


Figure 2.1: Example of a triangle inequality not verified

2.5 Experiments and results

In this Section, the experiments to evaluate the linear formulations are presented. The tests were performed on an Intel Xeron Gold 6240R 2.40GHz with 256GB of RAM memory. The commercial solver ILOG CPLEX Optimization Studio version 20.1.0 was used and a time limit of one hour was set for each instance.

Below, the results are divided into two parts. The first for the IRP and the second for the HIRP-BS.

2.5.1 Literature instances

The set from the literature instances introduced in Subsection 2.4.1 was chosen randomly to test the formulation and the results are presented in Table 2.6.

Table 2.6: Results for the classical IRP set

abs	$\vert\mathcal{N}\vert$	$\vert\mathcal{T}\vert$	#Vehicles	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$
5	30	3	1	9773.08	9773.90	34.84	0
5	30	3	2	10062.50	10063.50	1431.12	0
5	30	3	3	10321.70	10449.90	3600.00	1
5	50	3	1	15677.10	15678.70	187.85	0
5	50	3	2	15916.60	16123.60	3600.00	1
5	50	3	3	15688.90	16129.10	3600.00	3
2	40	3	1	11316.70	11317.80	395.22	0
2	40	3	2	11554.40	11689.00	3600.00	1
2	40	3	3	11802.30	12015.50	3600.00	2
Average						2227	0.89

Note that a very small gap of 0,89% is observed for these instances and that the average total time to solve is about 38 minutes as shown in Table 2.6. The results for the other literature set of instances are presented in Appendix B. Each table refers to 3 or 6 periods of time and contains from 1 and up to 5 vehicles. The lower and upper bounds, the time to obtain the solution (with one hour of time limit) and the gap are presented for each instance.

2.5.2 New benchmark instances

The new set of benchmark instances for the HIRP-BS introduced in Section 2.4.2 is composed of three groups: Small, Medium and Large. The results for each of them are presented below.

Table 2.7: Results for the small HIRP-BS instances set

Instances			CPLEX						
$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	ID	z_{RL}	$t_{RL}(s)$	z_{LB}	z_{UB}	$gap(\%)$	$t_{target}(s)$	$t_{total}(s)$
		1	14362.21	0.59	null	null	null	null	null
		2	14349.82	0.75	null	null	null	null	null
		3	13192.77	0.61	13468.21	13787.07	2.31	2755	3605.44
		4	15117.76	0.80	15410.03	16070.71	4.11	3602	3603.04
		5	12649.01	0.69	13048.71	14871.80	12.26	3606	3607.43
		6	12013.27	0.68	12365.08	12709.81	2.71	3605	3607.58
19	7	7	11091.07	0.47	11395.59	11667.71	2.33	3605	3606.29
		8	13417.14	0.85	13674.95	13948.05	1.96	3603	3603.39
		9	11195.59	0.55	11491.10	11722.88	1.98	3560	3603.90
		10	12406.95	0.73	12752.75	13327.03	4.31	3603	3604.07
		11	15287.84	0.80	15568.40	17630.62	11.70	3600	3600.14
		12	15118.76	0.74	15388.55	15689.12	1.92	3562	3608.62
		13	16058.63	0.77	16285.27	16535.68	1.51	3607	3609.31
						Average	4.28	3519	

In Table 2.7, for the two first instances considered, no results were found by CPLEX within the one hour of execution. For the others, a gap is always observed since no optimality is proven in less than one hour. The linear relaxation is easily solved for this group of instances. In Table 2.8, only the results for the linear relaxation can be reported since CPLEX is not able to provide any integer feasible solution for this group of medium-scale instances. The same occurs for the large-scale instances which the results are given by Table 2.9.

The results for the new benchmark instances are presented in Appendix C. Note that the tabled are divided according to the instances dimension (Small, Medium and Large) and the tables share the results with the metaheuristic of the next Chapter. Thus, only the first columns (CPLEX) have to be considered for the linear formulation.

Table 2.8: Results for the medium HIRP-BS instances set

Instances			CPLEX		Instances			CPLEX	
$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{T}\rvert$	ID	z_{RL}	$t_{RL}(s)$	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{T}\rvert$	ID	z_{RL}	$t_{RL}(s)$
19	14	1	31765.05	1.37	58	7	1	46315.26	24.86
		2	35995.55	1.14			2	27474.70	21.40
		3	36849.87	1.57			3	37506.26	18.19
	21	1	70705.51	1.77		14	1	100261.9994	32.61
		2	67279.76	1.58			2	102989.69	29.79
		3	81136.08	1.61			3	118288.86	20.56
	28	1	105572.10	2.73		21	1	199673.24	78.55
		2	108853.94	2.52			2	198632.41	65.69
		3	96870.65	2.50			3	187080.89	95.89
34	7	1	21504.21	4.03		28	1	312777.77	86.24
		2	25220.85	3.50			2	289651.41	134.67
		3	19551.36	3.08			3	344415.46	120.88
	14	1	64725.84	4.74	83	7	1	46176.09	192.97
		2	58026.86	8.71			2	45115.88	37.98
		3	60027.35	7.40			3	48785.67	151.18
	21	1	112811.33	14.25		14	1	150855.25	94.43
		2	142540.13	12.29			2	115554.70	332.01
		3	125961.64	16.06			3	161514.80	349.82
	28	1	199983.60	14.58		21	1	220407.99	405.76
		2	194899.21	18.19			2	272141.83	281.30
		3	198494.93	18.27			3	254559.92	546.52
46	7	1	28952.92	9.11		28	1	-	-
		2	31393.56	7.63			2	458910.42	618.15
		3	25408.63	4.75			3	-	-
	14	1	70858.715	14.24					
		2	85473.87	19.85					
		3	78769.42	13.58					
	21	1	163482.68	49.02					
		2	165078.23	47.82					
		3	129226.16	32.59					
	28	1	232277.48	56.34					
		2	263986.86	62.94					
		3	249555.83	59.98					

Table 2.9: Results for the large HIRP-BS instances set

Instances			CPLEX	
$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	ID	z_{RL}	$t_{RL}(s)$
114	7	1	63214.61	148.56
114	7	2	62803.88	92.41
114	7	3	62229.09	135.55
114	14	1	160846.19	313.28
114	14	2	240633.57	164.75
114	21	1	368429.18	1234.34
149	28	1	-	-
170	21	1	-	-
170	28	1	-	-
183	7	1	-	-

2.6 Chapter conclusion

This Chapter has presented the formal description and the mathematical formulation for both IRP and HIRP-BS. It includes the respective objective function, variables and constraints regarding the inventory level, the delivery quantities, the flow (degree), the vehicles capacity and the sub-tour elimination. For the IRP, the 800 classical literature instances was presented. For the HIRP-BS, since the problem is introduced in this thesis, a new set of instances was introduced and the generation method was explained considering each new feature comparing to the classical IRP.

Chapter remainder

- *Literature mathematical formulation for the classical IRP*
- *New mathematical formulation to address the HIRP-BS*
- *Computational experiments on both IRP and HIRP-BS*

Chapter 3

Iterative approach over periods

Go to the Table of Contents ↺

Abstract

In this chapter, a heuristic method that employs an iterative approach to tackle the Inventory Routing Problem (IRP) by decomposing it into subproblems based on the time horizon is presented. The proposed method systematically progresses through the time horizon, addressing each period in sequence from the first to the last. At each iteration, a subproblem is formulated, incorporating all concerned problem constraints but focusing only on a specific interval of the time horizon. The solution obtained from the previous periods serves as the starting point for solving the current subproblem. A key feature of this method is its ability to limit modifications to the solutions of earlier periods, which have already been optimized in previous iterations. It not only preserves the integrity of previous solutions but also significantly accelerates the resolution process, as the method avoids unnecessary recalculations and focuses computational efforts on refining the current period solution. The results of the experiments indicate that this approach is competitive in terms of both solution quality and execution time. It has demonstrated the ability to generate solutions for the set of benchmark instances considered, offering a practical and efficient method for solving IRP instances within a reasonable computational timeframe.

Chapter content

3.1 Chapter introduction 77
3.2 Iterative algorithm 78
3.3 Experiments 82
3.3.1 Instances 82
3.3.2 Results for the literature instances 83
3.4 Chapter conclusion 88

3.1 Chapter introduction

The iterative approach consists of a hybrid heuristic based on the formulation presented in Section 2.2 from Chapter 2. The method iterates over the time horizon, solving the subproblem composed by all constraints until the current period, and starting the resolution from the partial solution of the previous iteration. Furthermore, the number of modifications to the predefined partial solution is limited by a fixed method parameter.

The iterative algorithm as well as the problem and subproblem definitions are formally introduced in Section 3.2. To test the approach, Section 3.3 presents the different experiments conducted on the instances set (Subsection 3.3.1) and the results are presented and discussed in Subsection 3.3.2.

3.2 Iterative algorithm

Let $\mathcal{P}$ represent the IRP, and let $\mathcal{P}^t$, for all $t \in \mathcal{T}$, denote the t^{th} subproblem among the $|\mathcal{T}|$ possible subproblems. Solving $\mathcal{P}^t$ involves starting from the previously determined values for the route compositions, expressed by the variables $x^t_{i,j}$ from periods 1 to $t-1$, and then solving the current subproblem $\mathcal{P}^t$ while allowing a degree of freedom for the variables x during the execution of the algorithm for $\mathcal{P}^t$.

The overall concept is illustrated in Figure 3.1.

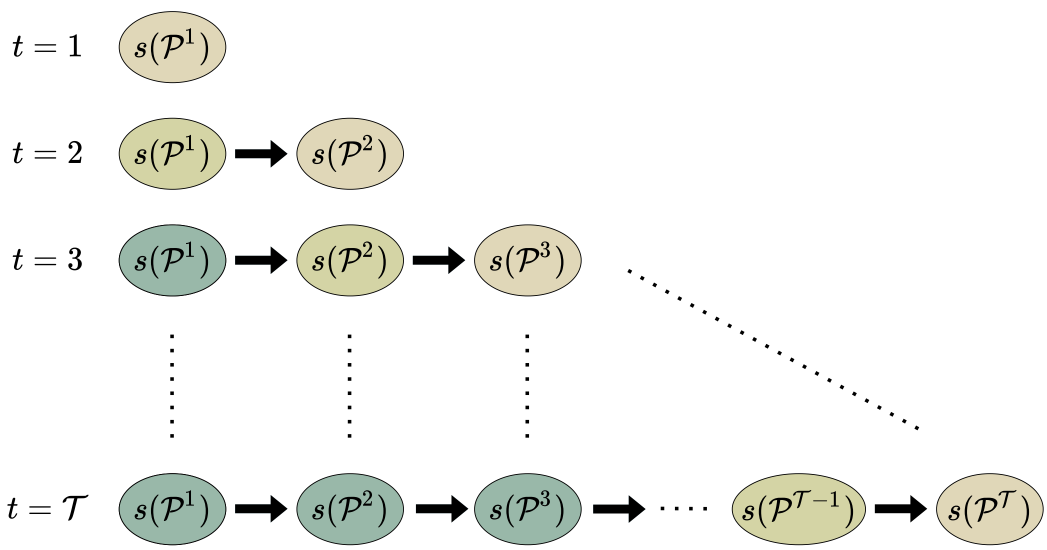


Figure 3.1: Iterative heuristic idea

In Figure 3.1, the interdependence among the subproblems for a given period t is depicted. For example, when $t = 2$, the subproblem $\mathcal{P}^2$ relies on the preceding subproblem $\mathcal{P}^1$, as it carries forward a partial solution that serves as the initial point for solving $\mathcal{P}^2$. This pattern of dependency continues as the time periods progress. When $t = 3$, the subproblem $\mathcal{P}^3$ draws information from $\mathcal{P}^2$, which itself depends on the solution obtained from $\mathcal{P}^1$. This dependency persists throughout the process, ensuring that each subproblem is informed by the solutions of its predecessors. The algorithm systematically iterates over the entire time horizon, progressing from one period to the next, and concludes only after all subproblems have been thoroughly explored.

It is also essential to highlight that as the algorithm progresses through the periods, the size and complexity of the subproblems increase. Specifically, the dimensions of the subproblems grow with each subsequent iteration, meaning that $|\mathcal{P}^1| < |\mathcal{P}^2| < \cdots < |\mathcal{P}^{\mathcal{T}-1}| < |\mathcal{P}^{\mathcal{T}}|$. This increasing complexity is presented in Figure 3.2, which illustrates how the subproblems expand as the algorithm advances through the different periods. The gradual expansion of the subproblems reflects the accumulation of information and increase the decisions that need to be taken as the algorithm approaches the final time period.

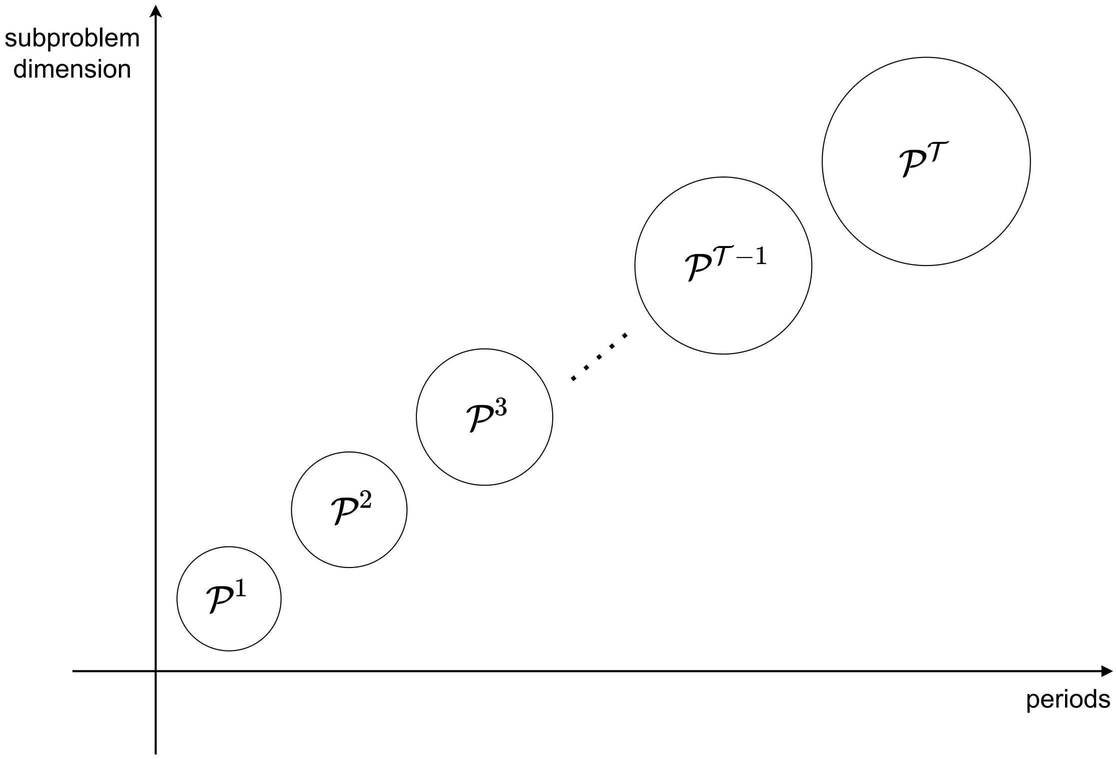


Figure 3.2: The iterative heuristic search space evolution

The underlying idea is to partially explore the neighborhood associated with each time period during each iteration. This exploration is guided by the variables that represent each period within the time horizon, allowing the solver to implicitly follow a logical sequence that aligns with the chronological order of the periods. It also guides the solver to know better the problem structure into periods since it may not be clear while considering all the variables separately. Such an approach is commonly utilized in constraint programming algorithms, as discussed in previous works Bourreau et al. (2019, 2020).

To implement this approach, for a given period $t \in \mathcal{T}$, the variables $\bar{x}^t_{i,j}$, which represent the values obtained from solving the subproblem $\mathcal{P}^{t-1}$, are considered. The next step involves calculating the distance between these variables and their corresponding counterparts $x^t_{i,j}$ in the subproblem $\mathcal{P}^t$. This calculation is done using Equations (3.1). Consequently, the non-linear nature of Equations (18) is addressed by linearizing them through Constraints (3.2). The linearization process employed here is consistent with the method that is presented in Chapter 4, Section 4.5, which corresponds to the post-optimization phase of the SEMPO metaheuristic. This step is crucial as it ensures the compatibility of the equations within a linear model, facilitating the iterative solution of the subproblems across different periods.

$$\delta^{t'}_{i,j} = |\bar{x}^{t'}_{i,j} - x^{t'}_{i,j}| \qquad \forall i,j \in \mathcal{N}', t' \in \{1,...,t-1\} \tag{3.1}$$

$$\eta = \begin{cases} \delta_{i,j}^{t'} \geq \bar{x}_{i,j}^{t'} - x_{i,j}^{t'} \\ \delta_{i,j}^{t'} \geq x_{i,j}^{t'} - \bar{x}_{i,j}^{t'} \\ \delta_{i,j}^{t'} \leq \bar{x}_{i,j}^{t'} + x_{i,j}^{t'} \\ \delta_{i,j}^{t'} \leq 2 - \bar{x}_{i,j}^{t'} - x_{ij}^{t'} \end{cases} \quad \forall i, j \in \mathcal{N}', t' \in \{1, ..., t-1\} \tag{3.2}$$

Only the x variables are considered due to the fact that deactivating ($x_{i,j}^t = 0$) or activating ($x_{i,j}^t = 1$) an arc in a given period directly corresponds to either completely removing or adding a quantity q at a customer. This relation is already accounted for by the problem constraints, which establish the connection between the x and q variables (see Constraints (2.8) from the linear formulation of the classical IRP presented in Section 2.2 of Chapter 2). By focusing on the x variables, the changes in customer deliveries are effectively managed since these deliveries are tied to the activation or deactivation of arcs.

Once the resolution of $\mathcal{P}^t$ begins, starting from the known values of the x variables up until period $t-1$, the corresponding q values can be more easily determined based on whether an arc is utilized in a route or not. However, during the transition from subproblem $\mathcal{P}^{t-1}$ to $\mathcal{P}^t$, an initial value for the q variables is set without allowing any degree of freedom, thereby guiding the search process.

Following this, the degree of freedom for the x variables is expressed by

$$\sum_{(i,j)\in\mathcal{A}} \sum_{t'\in\{1,...,t-1\}} \delta_{i,j}^{t'} \leq \left\lceil \hat{x} \cdot \frac{\Delta}{100} \right\rceil \tag{3.3}$$

in which $\hat{x}$ represents the number of non-zero variables $x_{i,j}^t$ obtained from solving the subproblem $\mathcal{P}^{t-1}$, i.e., all variables where $x_{i,j}^t = 1$. The parameter Δ denotes the percentage of allowed changes in each subproblem $\mathcal{P}^t$ based on previously activated arcs. Specifically, when $\Delta = 0$, no changes to the previously activated arcs are permitted, while when $\Delta = 100$, all previously activated arcs could potentially be altered.

The iterative approach is outlined in Algorithm 2.

Algorithm 2 starts by defining two essential sets, denoted as $\bar{X}$ and $\bar{Q}$, which are initially set to be empty (as outlined in line 1). Alongside this, the algorithm also initializes the elapsed time variable, $elpTime$, to zero, marking the start of the timing process (as described in line 2).

Following this, the algorithm proceeds to generate the first subproblem, $\mathcal{P}^1$, which is constructed using only the relevant data for the first period of the problem time horizon (as specified in line 3). This initial subproblem is then solved using a Mixed-Integer Linear Programming (MILP) solver, which applies the mathematical model defined by equations (2.1) through (2.18) (as detailed in Chapter 2, Section 2.2). The process of solving this first subproblem is critical, as it sets the stage for subsequent iterations (as indicated in line 4). Upon obtaining a solution for $\mathcal{P}^1$, the algorithm updates the elapsed time, $elpTime$, to the duration spent in solving this initial subproblem (line 5).

If a feasible solution is found for the first subproblem, the algorithm then moves forward by incorporating the associated decision variables x and q into the previously defined sets, $\bar{X}$ and $\bar{Q}$, respectively (this step is carried out between lines 6 and 23, specifically in lines 8 and 9). The inclusion of these variables into the sets $\bar{X}$ and $\bar{Q}$ is a crucial as it allows the algorithm to reuse these values in subsequent periods.

As the algorithm transitions to the second period and continues through to the final period of the time horizon, it enters a loop (as delineated between lines 10 and 23). Within each iteration of this loop, the algorithm integrates the solution obtained from

Algorithm 2: Iterative algorithm

Data: problem *data* from Chapter 2, Section 2.2
degree of freedom Δ
solver time limit $timeMax$

Result: a feasible solution $s(\mathcal{P}^{\mathcal{T}})$ for the $\mathcal{P}^t$ subproblem

1 $\bar{X}, \bar{Q} \leftarrow \varnothing$
2 $elpTime \leftarrow 0$
3 Create the first subproblem $\mathcal{P}^1$
4 $s(\mathcal{P}^1) \leftarrow$ solve($data$, $\mathcal{P}^1$, $timeMax$)
5 Update elapsed time $elpTime$
6 **if** $s(\mathcal{P}^1)$ is feasible **then**
7     $\bar{x}^1_{i,j}, \bar{q}^1_i \leftarrow$ getValues($s(\mathcal{P}^1)$)
8     $\bar{X} \leftarrow \bar{x}^1_{i,j}$
9     $\bar{Q} \leftarrow \bar{q}^1_i$
10   **for** each period $t \in \mathcal{T} \setminus \{1\}$ **do**
11   Create the t^{th} subproblem $\mathcal{P}^t$
12         $\mathcal{P}^t \leftarrow$ addHint($\bar{X}, \bar{Q}, \eta, \Delta$)
13         $timeMax \leftarrow timeMax - elpTime$
14         $s(\mathcal{P}^t) \leftarrow$ solve($data$, $\mathcal{P}^t$, $timeMax$)
15   Update elapsed time $elpTime$
16   **if** $s(\mathcal{P}^t)$ is feasible **then**
17             $\bar{x}^t_{i,j}, \bar{q}^t_i \leftarrow$ getValues($s(\mathcal{P}^t)$)
18             $\bar{X} \leftarrow \bar{X} \cup \bar{x}^t_{i,j}$
19             $\bar{Q} \leftarrow \bar{Q} \cup \bar{q}^t_i$
20   **else**
21   stop
22   **end**
23   **end**
24 **else**
25   stop
26 **end**
27 **return** $s(\mathcal{P}^H)$

the previous period into the current subproblem. Specifically, during the t^{th} iteration, the algorithm constructs the t^{th} subproblem (as indicated in line 11). Since the solution from the preceding period, $t-1$, is now available, additional constraints are introduced into the t^{th} subproblem to guide the search for an optimal solution. These constraints are formulated by incorporating the Δ ratio, as well as the set of linear constraints η, which are represented by Inequalities 4.39 (as specified in line 12). The role of these constraints is to restrict the x variables within the allowable flexibility defined by Δ and to provide an initial value for the q variables, thereby improving the efficiency of the solution process.

As the loop progresses, the algorithm continuously updates the maximum allowable time, *timeMax*, based on the cumulative elapsed time from the previous iterations (as described in line 13). Subsequently, the algorithm solves the t^{th} subproblem and updates the elapsed time, *elpTime*, to the time spent in solving this particular subproblem (this update is captured in line 15). If the algorithm succeeds in finding a feasible solution (this check occurs between lines 16 and 19), it extracts the solution (as detailed in line 17) and adds the corresponding values to the sets $\bar{X}$ (line 18) and $\bar{Q}$ (line 19), thereby incrementally building the final solution.

Lastly, after all iterations have been completed and each subproblem has been addressed, the algorithm returns the solution obtained from the last subproblem, $\mathcal{P}^H$ (as indicated in line 27). This final step concludes the algorithm execution, providing the complete solution to the problem.

3.3 Experiments

The experiments were conducted on a high-performance computing environment equipped with an AMD EPYC 7452 32-Core processor, supported by 512GB of RAM. The implementation was carried out using the C++ programming language. For solving the complex MILP model, the solver IBM ILOG CPLEX Optimization Studio, specifically version 22.1.1.0, was chosen.

The experimental setup focused on the set of classical IRP instances, originally proposed by Archetti et al. (2007); Coelho et al. (2012a). These instances serve as a benchmark in the field, providing a standardized way to evaluate and compare the performance of different algorithms. In addition to these classical instances, a series of experiments were conducted to explore the impact of varying the maximum authorized distance for the x variables. The results of these variations are presented in the subsequent sections, offering insights into how different constraints on x influence the overall solution quality and computational efficiency.

For each IRP instance and the corresponding method applied, the MILP solver was given a strict time limit of one hour to find a solution. In the case of the iterative method, this time constraint is cumulative across periods; that is, when solving an instance up to a certain period t, the elapsed time is the sum of the times spent from period 1 to period t. This is due to the fact that the subproblem $\mathcal{P}^t$ in the iterative method reuses information from previously solved subproblems, making it period-dependent. Conversely, in the classical approach, the time limit is applied individually to each instance, as there is no inter-period dependency in solving the subproblems. This distinction highlights the iterative method increasing complexity.

Furthermore, the parameter Δ was systematically varied across a predefined set of values, specifically $\Delta = \{0, 10, 20, \dots, 90, 100\}$, to thoroughly analyze its impact on the algorithm performance in solving the problem. These Δ values, representing different levels of tolerance or flexibility within the model, were held constant throughout the execution of the algorithm, allowing for a clear comparison of their effects on the solution process. The choice of these particular values was made to cover a wide range of possible scenarios, providing a comprehensive understanding of how varying Δ influences the efficiency and effectiveness of the solution approach.

3.3.1 Instances

To evaluate the performance of the iterative algorithm, the well-known set of classical IRP instances was utilized. The detailed description and characteristics of these specific IRP instances were thoroughly introduced in Chapter 2, Subsection 2.4.1.

3.3.2 Results for the literature instances

Table 3.1 presents the results for the same group of instances chosen on the previous chapters. Columns *abs*, $|\mathcal{N}|$, $|\mathcal{T}|$ and *vehic* present the instances characteristics. The next four columns z_{LB}, z_{UB}, $t(s)$, $gap(\%)$ present the results for the lower and upper bounds, time in seconds and the gap for the iterative algorithm. The next four columns (also z_{LB}, z_{UB}, $t(s)$, $gap(\%)$) consider the results for the linear formulation on its classical version without the iterative approach. The last column $globalGap(\%)$ compares the gap between both approaches (iterative and classical).

Table 3.1: Results for the classical IRP instances

				Iterative algorithm				Classical formulation				
abs	$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	#Vehicles	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	$globalGap(\%)$
5	30	3	1	10320.60	10321.60	84.02	0	9773.08	9773.90	34.84	0	5
5	30	3	2	10549.10	10550.10	2315.07	0	10062.50	10063.50	1431.12	0	5
5	30	3	3	11024.20	11025.30	2732.16	0	10321.70	10449.90	3600.00	1	5
5	50	3	1	16174.30	16175.90	383.72	0	15677.10	15678.70	187.85	0	3
5	50	3	2	16588.20	16688.80	3600.00	1	15916.60	16123.60	3600.00	1	3
5	50	3	3	16828.10	17208.20	3600.00	2	15688.90	16129.10	3600.00	3	6
2	40	3	1	11795.70	11796.80	432.17	0	11316.70	11317.80	395.22	0	4
2	40	3	2	12126.30	12127.50	222.02	0	11554.40	11689.00	3600.00	1	4
2	40	3	3	12480.50	12481.80	2548.52	0	11802.30	12015.50	3600.00	2	4
			Average			1768	0.34			2227	0.89	4

Note that in Table 3.1 the average time for the iterative approach is of about 30 minutes while the classical approach is 38 minutes. When comparing both approaches, the last column show an average gap of 4%, which shows that the iterative approach is competitive to provide good results for these instances. In the following, results for other classical IRP instances are presented and discussed.

Tables 3.2 to 3.5 show the results for a particular subset of instances where the parameter Δ is set to 50, and involve either 2 or 3 vehicles. The delta value was chosen to be 50 since it presents the best results considering the other ones from 0 to 100. These specific instances were chosen based on their representativity among the whole set. In these tables, a detailed comparison is provided against the benchmark values reported by Manousakis et al. (2021), who have established the so far best-known bounds for these classical instances in the literature. This comparison highlights the relative performance of the current algorithm against the state of the art, offering insights into its effectiveness and efficiency within this context.

The results presented in Table 3.2 demonstrate that, on average, the iterative method is ten times faster than direct resolution, with an average solution gap of 9%. However, a closer, instance-by-instance analysis reveals significant variability in performance. Specifically, five instances − low abs1n20, low abs1n25, high abs1n15, high abs1n20, and high abs1n25 − exhibit a tenfold acceleration in computation time. Among these, the instance abs1n25.dat shows a 19% gap, highlighting a trade-off between speed and solution quality in some cases. The least favorable is observed for the instance low abs1n5.dat, with a time horizon of three periods, where a 22% cost gap and a computation time ratio of 1 are observed. On the other hand, the most favorable instance is high abs1n20.dat, also with a three-period time horizon, which shows only a 4% cost gap and a computation time ratio of 39.

In Tables 3.3 and 3.5, a comparison of the iterative approach with the best-known upper bounds from the literature, as provided by Manousakis et al. (2021), reveals that

Table 3.2: Results for $\Delta = 50$ and 3 vehicles

instance				Iterative heuristic		MILP model		$gap(\%)$	$ratioTime$
type	name	$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	UB	$time(s)$	UB	$time(s)$		
low	abs1n5	5	3	1826.68	0	1430.51	0	22	2
low	abs1n10	10	3	2894.35	1	2732.61	5	6	3
low	abs1n15	15	3	3073.35	9	2783.77	86	9	10
low	abs1n20	20	3	3913.45	139	3605.72	3600	8	26
low	abs1n25	25	3	4312.25	107	3503.30	3600	19	34
low	abs1n30	30	3	4764.89	406	4251.64	3600	11	9
low	abs1n35	35	3	4836.19	3596	4099.58	3600	15	1
low	abs1n40	40	3	5367.35	3593	4552.06	3600	15	1
low	abs1n45	45	3	5309.61	1782	4537.30	3600	15	2
low	abs1n50	50	3	5769.00	2046	5442.62	3600	6	2
high	abs1n5	5	3	2695.22	0	2298.73	0	15	1
high	abs1n10	10	3	5666.74	1	5506.09	6	3	3
high	abs1n15	15	3	6480.80	9	6242.90	104	4	11
high	abs1n20	20	3	8506.00	85	8165.42	3600	4	42
high	abs1n25	25	3	9711.60	71	8893.88	3600	8	51
high	abs1n30	30	3	13464.80	833	12908.90	3600	4	4
high	abs1n35	35	3	13125.20	2459	12445.00	3600	5	1
high	abs1n40	40	3	15037.00	3591	14224.10	3600	5	1
high	abs1n45	45	3	15443.10	1466	14771.00	3600	4	2
high	abs1n50	50	3	16292.90	1331	15926.00	3600	2	3
							Average	9	10

Table 3.3: Results for $\Delta = 50$ and 3 vehicles

instance				Iterative heuristic		Manousakis et al. (2021)		$gap(\%)$	$ratioTime$
type	name	$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	UB	$time(s)$	UB	$time(s)$		
low	abs1n5	5	3	1826.68	0	1430.51	1	22	1
low	abs1n10	10	3	2894.35	2	2732.61	21	6	11
low	abs1n15	15	3	3073.35	9	2783.77	23	9	3
low	abs1n20	20	3	3913.45	139	3605.72	196	8	1
low	abs1n25	25	3	4312.25	107	3503.38	83	19	1
low	abs1n30	30	3	4764.89	407	4251.64	1069	11	3
low	abs1n35	35	3	4836.19	3596	4080.60	2463	16	1
low	abs1n40	40	3	5367.35	3593	4532.84	13369	16	4
low	abs1n45	45	3	5309.61	1783	4537.30	29437	15	17
low	abs1n50	50	3	5769.00	2046	6017.66	42832	-4	21
high	abs1n5	5	3	2695.22	0	2298.73	0	15	0
high	abs1n10	10	3	5666.74	2	5506.09	13	3	8
high	abs1n15	15	3	6480.80	9	6242.90	16	4	2
high	abs1n20	20	3	8506.00	85	8165.42	229	4	3
high	abs1n25	25	3	9711.60	71	8893.82	53	8	1
high	abs1n30	30	3	13464.80	834	12098.90	2409	10	3
high	abs1n35	35	3	13125.20	2460	12396.00	3380	6	1
high	abs1n40	40	3	15037.00	3592	14224.10	9173	5	3
high	abs1n45	45	3	15443.10	1467	14771.00	34500	4	24
high	abs1n50	50	3	16292.90	1331	16115.80	43101	1	32
							Average	9	7

Table 3.4: Results for $\Delta = 50$ and 2 vehicles

instance				Iterative heuristic		MILP model		*gap*(%)	*ratioTime*
type	name	$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	*cost*	*time*(*s*)	*cost*	*time*(*s*)		
low	abs1n5	5	6	3776.60	1	3775.80	5	0	7
low	abs1n20	20	6	7392.36	2791	7388.92	3600	0	1
low	abs1n25	25	6	8233.73	3329	7464.18	3600	9	1
high	abs1n5	5	6	6380.03	1	6380.03	3	0	5
high	abs1n15	15	6	13105.10	126	12624.70	3600	4	29
high	abs1n20	20	6	15585.80	3053	15553.20	3600	0	1
high	abs1n25	25	6	16218.50	3062	15958.40	3600	2	1
							Average	2	6

Table 3.5: Results for $\Delta = 50$ and 2 vehicles

instance				Iterative heuristic		Manousakis et al. (2021)		*gap*(%)	*ratioTime*
type	name	$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	*cost*	*time*(*s*)	*cost*	*time*(*s*)		
low	abs1n5	5	6	3776.60	1	3775.68	8	0	11
low	abs1n20	20	6	7392.36	2791	7388.80	2276	0	1
low	abs1n25	25	6	8233.73	3329	7461.55	735	9	0
high	abs1n5	5	6	6380.03	1	6379.56	8	0	13
high	abs1n15	15	6	13105.10	126	12624.70	148	4	1
high	abs1n20	20	6	15585.80	3053	15540.40	3232	0	1
high	abs1n25	25	6	16218.50	3062	15954.80	197	2	0
							Average	2	4

the iterative method is competitive in terms of time convergence, despite some gaps being observed in certain cases.

Tables 3.6 and 3.7 summarize the results for classical IRP instances over time horizons of 3 and 6 periods, respectively. The analysis focuses exclusively on the first type of instance ($abs1nB.dat$). In these tables, the column labeled *type* indicates the type of inventory holding costs considered, $gapX(\%)$ represents the average gap across all instances up to the X^{th} period, *timeRatio* denotes the ratio between the time required to solve the last period using the classical approach and the cumulative time for the iterative heuristic, and finally, σ provides the standard deviation of the *timeRatio* values for a specified Δ.

In Tables 3.6 and 3.7, it is important to note that *gap* values can only be calculated starting from the second period. This is because, in the first period, both the iterative and classical approaches yield identical results, as there is no prior information available to incorporate into the first subproblem in the iterative method. Consequently, the initial period does not provide a basis for calculating a performance gap between the two methods.

During the first period, for the majority of instances analyzed, no routes are scheduled because customers have sufficient inventory levels, meaning that no product deliveries are necessary. As a result, there are no differences between the methods at this stage. Moving into the second period, all *gap* values are observed to be zero. This is attributed to the problem in this period, where the information carried over from the first period significantly aids in resolving the subproblem, ensuring that both methods arrive at the same solution.

However, as we progress to the third period and beyond, the complexity of the problem increases considerably. From $gap3$ onward, the problem dimension can expand significantly, making it more challenging to obtain optimal solutions. This is where the iterative method

Table 3.6: Results for $|\mathcal{T}| = 3$

type	Δ	*gap*2(%)	*gap*3(%)	*timeRatio*	σ
low	0	0	20	100.69	202.33
	10	0	19	61.06	168.77
	20	0	17	18.45	40.07
	30	0	13	6.64	19.54
	40	0	15	5.77	16.11
	50	0	12	5.22	15.86
	60	0	12	4.81	15.22
	70	0	12	4.56	15.11
	80	0	11	5.00	16.80
	90	0	10	4.89	17.52
	100	0	10	8.08	25.00
high	0	0	8	78.49	204.18
	10	0	8	57.29	239.40
	20	0	7	22.55	74.27
	30	0	5	5.73	14.85
	40	0	5	4.71	11.19
	50	0	5	4.13	9.30
	60	0	5	3.87	9.04
	70	0	4	3.78	9.53
	80	0	4	3.86	10.80
	90	0	4	4.04	11.66
	100	0	4	5.66	9.18
	avg	0	10	19.06	52.53

Table 3.7: Results for $|\mathcal{T}| = 6$

type	Δ	*gap*2(%)	*gap*3(%)	*gap*4(%)	*gap*5(%)	*gap*6(%)	*timeRatio*	σ
low	0	0	18	21	17	18	402.81	491.65
	10	0	19	12	12	10	36.17	32.06
	20	0	17	11	7	8	15.01	11.51
	30	0	16	8	5	5	11.13	10.97
	40	0	15	6	2	6	4.37	3.58
	50	0	15	0	7	-2	3.94	3.64
	60	0	14	-1	6	-4	2.77	2.93
	70	0	12	-1	4	-4	1.72	1.27
	80	0	12	-3	-1	-2	2.03	1.37
	90	0	10	-4	2	-6	1.57	0.67
	100	0	9	-4	-1	-7	1.43	0.84
high	0	0	7	8	11	11	692.68	1264.34
	10	0	8	4	6	5	33.75	43.56
	20	0	4	1	5	15	17.13	28.21
	30	0	7	2	1	0	8.14	11.55
	40	0	6	-1	4	-2	4.83	6.63
	50	0	6	-1	3	-2	5.43	8.06
	60	0	5	-1	2	-5	3.60	6.88
	70	0	4	-2	2	-5	2.66	4.20
	80	0	4	-1	1	-4	2.08	2.41
	90	0	3	-4	0	-5	1.61	2.11
	100	0	3	-5	-2	-5	1.23	1.42
	avg	0	10	2	4	1	57.09	88.18

starts to diverge in performance from the classical approach, as the problem becomes harder to solve with each subsequent period.

Regardless of the instance type or its size, a trend is observed: as the Δ value increases, there is a notable reduction in the gap. This trend occurs because a higher Δ value allows the subproblem more flexibility in adjusting the arcs, thereby increasing the chances of finding better solutions. However, this benefit comes at a cost-computation time increases as the search space expands with the larger Δ values. Despite this, even in the worst-case scenarios regarding time, the iterative approach generally outperforms the classical method on average, highlighting its efficiency in balancing solution quality and computational effort.

3.4 Chapter conclusion

In this chapter, an iterative approach for solving the IRP was introduced. This method integrates exact resolution techniques, based on a MILP formulation of the IRP, with an iterative exploration of the search space across the available time periods. The approach is structured so that, at each iteration, the subproblem corresponding to the current time period is formulated and solved, building upon the solution obtained from the preceding subproblem. This iterative process allows the method to refine and improve the solution progressively as it advances through the time periods.

The experiments were conducted using a set of classical instances from the literature, which serve as benchmarks in the field. The results demonstrated that this iterative method offers a significant competitive advantage, particularly when evaluating execution time and the solution quality within a one-hour time limit. The method presents a computational efficiency and the quality of the solutions, as indicated by the results.

Chapter remainder

- *An iterative algorithm to solve the IRP*
- *Decomposition method according to the number of periods available*
- *Sequential resolution from the first to the last period*
- *Partial dependance on the subproblems to accelerate the algorithm convergence*

Chapter 4

A Split-embedded Metaheuristic with a Post-optimization phase

Go to the Table of Contents ↺

Abstract

A Split-based metaheuristic is introduced in this Chapter to solve the IRP and the HIRP-BS. A Greedy Randomized Adaptive Search Procedure (GRASP) coupled with an Evolutionary Local Search (ELS) and a Post-optimization phase comprises the metaheuristic. A Split algorithm is the main core of the metaheuristic and relies on the definition of a multi-period giant tour containing the customers and predefined calculated quantities to be delivered and consists in splitting this sequence to define the routes that are assigned to the set of available vehicles. Once the search has reached the end of the giant tour, the shortest path retrieval is performed in order to retrieve the minimum-cost solution value found. The metaheuristic combines a three-step constructive heuristic capable of generating a good quality starting solution to be further improved by the ELS mechanism which includes a mutation, an evaluation and a local search phases. Lastly, the best solution obtained so far is improved in a Post-optimization phase that acts on the mathematical formulation introduced in the previous Chapter. Computational results are then performed in both literature and new instances sets and results are reported and discussed.

Chapter content

4.1 Chapter introduction 90
4.2 The SEMPO schema 90
4.3 Giant tour generation and Split algorithm 92
4.3.1 Generation of the giant tour 94
4.3.2 Splitting the giant tour 99
4.3.3 Shortest path retrieval 107
4.4 Evolutionary Local Search algorithm 108
4.4.1 Mutation 110
4.4.2 Evaluation 111
4.4.3 Local search 111
4.5 Post-optimization 119
4.5.1 Linearization constraints 119
4.5.2 Distance definition 119

4.6 Experiments and results 120
4.6.1 Results for the literature instances 121
4.6.2 Results for the new benchmark instances 122
4.7 Chapter conclusion 129

4.1 Chapter introduction

In this Chapter, a metaheuristic named Split-Embedded Metaheuristic with a Post-Optimization phase (SEMPO) is introduced to solve the HIRP-BS. The algorithm contains three main steps: a constructive heuristic, an Evolutionary Local Search (ELS) and a Post-Optimization step. The Split algorithm, which is the main step of the constructive heuristic, is presented in Section 4.3 and considers the decomposition of a multi-period giant tour into routes with a shortest path with ressource consumption algorithm to create feasible solutions for the problem. The objective is to generate a good quality solution to be further improved.

The ELS step is introduced in Section 4.4 as well as the mutation and local search mechanisms. Then, an original post-optimization phase is presented in Section 4.5 to try to improve the solutions even more after the ELS. Experiments are conducted on the new set of instances proposed in this thesis and the results are presented in Section 4.6.

4.2 The SEMPO schema

The Split-based metaheuristic (SEMPO) integrates a Greedy Randomized Adaptive Search Procedure (GRASP) algorithm that utilizes the Split mechanism at various stages of its process, in combination with an Evolutionary Local Search (ELS) and a Post-Optimization phase, which includes a local search as illustrated in Figure 4.1. The GRASP metaheuristic is a well-known approach that iteratively constructs initial solutions and applies a local search procedure until a predefined stopping criterion is reached. Numerous researchers have applied this method to VRP problems (Kontoravdis and Bard 1995, Villegas et al. 2011, Guemri et al. 2016) as well as ELS algorithms (Duhamel et al. 2011, Zhang et al. 2015). While GRASP and ELS have been widely explored for VRP, their application specifically to the IRP has been less studied, with no recent publications identified.

Therefore, the metaheuristic contains the following three main steps:

Step 1. A three-step constructive heuristic that provides feasible solutions even for large-scale instances. These are the three steps:

1. Definition of a multi-period giant tour containing a sequence of customers to be visited and the quantities to be delivered that favor the split decomposition into routes of reasonable quality. It relies on the possibility of anticipating the customer deliveries to previous time periods.

2. Evaluation of the previous generated giant tour in order to decompose the sequence into routes that are assigned to the set of available vehicles at each period of time. This evaluation step is obtained thanks to the Split algorithm.

3. Improvement of the solution obtained so far by a local search mechanism that acts on the routing part of the problem by four neighborhoods considering intra and inter

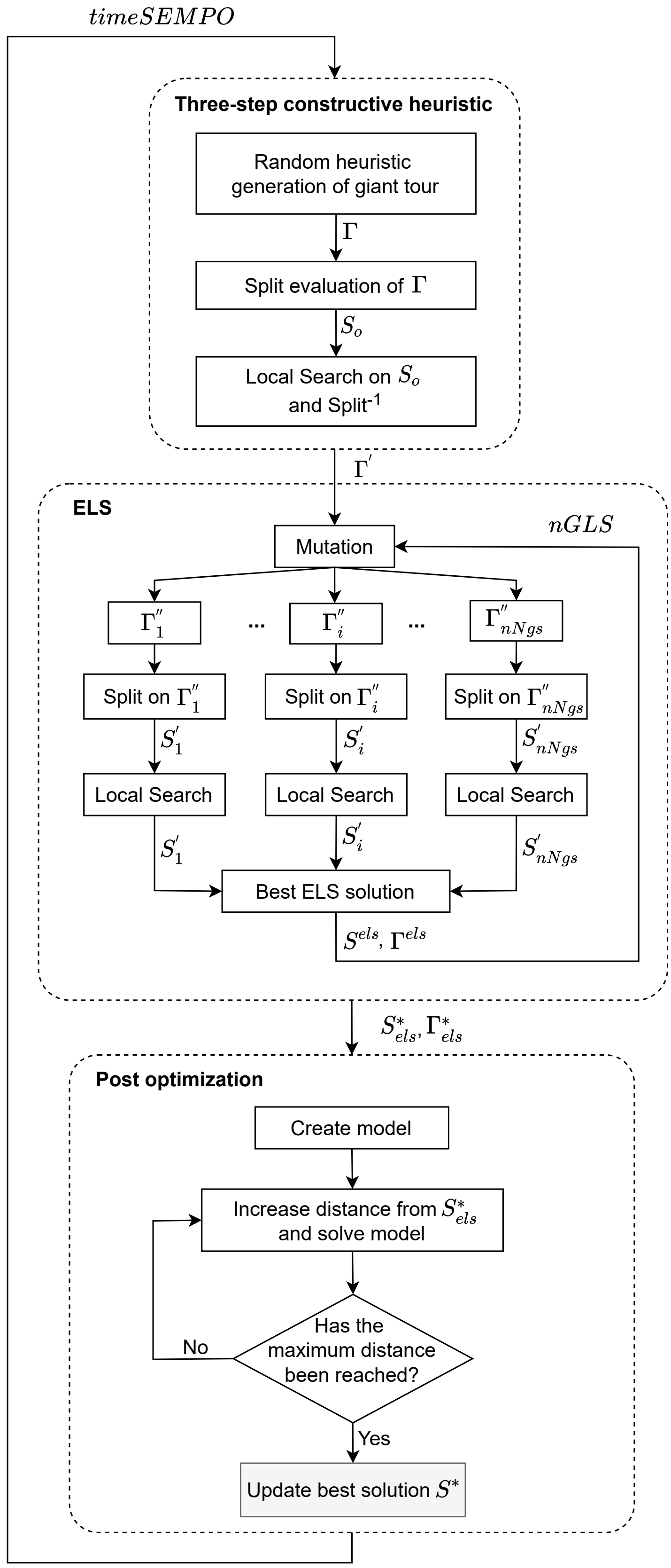


Figure 4.1: Schema of the SEMPO algorithm

routes moves. Once done, the improved solution is converted into a giant tour to restart the process if needed.

Step 2. An Evolutionary Local Search (ELS) procedure to improve the solution obtained in the previous stage for a given number of iterations without improvement. The ELS combines also three phases as follows:

1. A mutation phase that provides diversification by permuting the giant tour positions among those belonging to the same period of time.
2. An evaluation phase which corresponds to the algorithme Split and
3. A local search mechanism that are the same from the previous three-step constructive heuristic presented above.

Step 3. The Post-Optimization procedure seeks to enhance the solution by refining the best ELS outcome solution obtained in the current GRASP iteration. This method operates on the premise that the search space of the mathematical model can be constrained by defining a "distance" from the best ELS solution. By narrowing the search space, this approach reduces the computational effort required to solve an MILP model while focusing on areas with a higher probability of improvement. During this stage, the delivery quantities and route configurations are optimized. The model builds on the mathematical formulation introduced in Section 2.3 of Chapter 2 by adding constraints related to the "distance" from the ELS solution and targeting solutions with routes that remain "close" to those identified in the previous stage.

The initial two steps of the proposed constructive heuristic, which involve generating a giant tour by determining the quantities and positioning of customers in Γ, followed by creating a feasible solution for the HIRP-BS using the Split algorithm, are detailed in Section 4.3. The ELS algorithm, which includes the local search procedure also used in the third step of the constructive heuristic, is discussed in Section 4.4. Finally, the Post-Optimization procedure is described in Section 4.5.

4.3 Giant tour generation and Split algorithm

Given a sequence of customers, the Split algorithm for the VRP focuses on finding the shortest path in a graph. For the capacitated version (CVRP), it not only seeks the shortest path but also accounts for resource constraints due to the limited number of vehicles. In the case of the IRP, the Split algorithm further incorporates inventory management into the process of route creation. The primary objective of the algorithm is to efficiently solve the combined challenges of routing and inventory management.

A schema of the Split algorithm is presented in Figure 4.2.

In Figure 4.2, customers A, B, C, D, and E are scheduled over a 3-period horizon and arranged into a giant tour based on their product needs, considering a scenario with no previous delivery operations, only their initial inventory level s_i. The arc from node 0 to 1 represents the route $0 \to C \to 0$, which delivers q_c^1 units of products during period 1. This scenario is similar to the one presented by Prins (2004) for the VRP.

The originality of the approach introduced in this thesis is illustrated by two particular cases: anticipating a delivery from one customer assigned to a period t to a period t' such that $t' < t$ and that the customer (i) has not already been treated in the current route and (ii) has already been treated in the current route. These two cases are illustrated below.

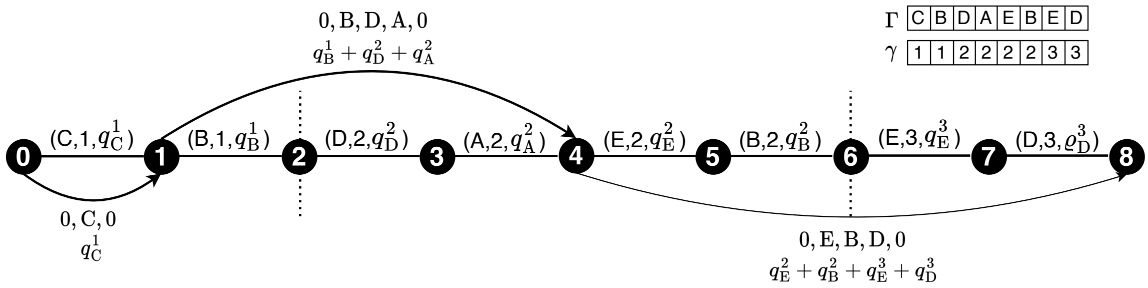


Figure 4.2: Split algorithm general idea

(i) The second route {0, B, D, A, 0} represents an anticipation delivery scenario since customers D and A are initially scheduled at period 2, but the route is assigned to period 1 given the time period of the first customer B. Since a constructive heuristic to define the giant tour places the quantities at the very latest possible period, assigning this route to period 2 would cause an inventory disruption since the quantity q^1_B is supposed to be delivered at period 1 at most

(ii) The third rote given by the sequence {0, E, B, D, 0} also anticipates the deliveries and has the particularity of accumulating the quantity to be delivered to customer E at period 3 with those of period 2, and also the fact that this customer only appears once on the route, which may reduce transportation costs whereas the inventory costs increase. This trade-off has been explained in Chapter 1, Subsection 1.3.4, when defining a feasible IRP solution.

These particular cases differ from the classical Split for the VRP and variations and can handle the transportation and the inventory problem simultaneously. More specifically, the algorithm consists of the following steps:

1. **Step 1: Defining of a giant tour.** Generate a multi-period giant tour containing a sequence of customers and predefined quantities to be delivered and a number of periods which correspond to an estimation of the latest possible moment to deliver the given quantity. This placement is done by a constructive heuristic that allows finding the "best" moment in which the customer should be placed and the quantity and possible period to be assigned.

2. **Step 2: Splitting the giant tour.** From the left to the right, splitting the sequence taking into account the problem constraints given in Section 2.3 by Constraints (2.23) to (2.40) including the customers storage capacity, the inventory level and the capacity of the vehicles (heterogeneous fleet). It proceeds to an exploration of the possible routes to be created by adding labels that contains information on the candidate route. A limit must be imposed in order to reduce the space complexity since one label may generate multiple others. To select the best candidates, the labels are sorted by their increasing solution costs. Once the last customer in the sequence is chosen, the procedure ends.

3. **Step 3: Retrieving the critical path.** From the right to the left, this step aims to retrieve the critical path that corresponds to the minimum cost solution obtained once the giant tour exploration is done by the previous step. Picking any label on

the last customer on the sequence allows finding a feasible solution for the problem. It can be done by retrieving the customers that originates the route as well as the vehicle and the period assigned since the parent nodes and indices were saved through the algorithm execution. This step can be seen as an application of a shortest path algorithm.

To the best of my knowledge, no existing literature addresses a Split-based algorithm for inventory routing problems that considers a multi-period giant tour. The process of defining the giant tour is detailed in Subsection 4.3.1, the splitting phase is explained in Subsection 4.3.2, and the shortest path retrieval for identifying a feasible solution is covered in Subsection 4.3.3.

4.3.1 Generation of the giant tour

A giant tour is defined on an auxiliary directed graph $G = (\mathcal{V}', \mathcal{A})$. The vertex subset $\mathcal{V}$ represents the customers, and $\mathcal{V}' = \mathcal{V} \cap 0$ includes both the supplier and the customers. For routing problems, each node in $\mathcal{V}$ represents a customer that must be visited by a vehicle. In the IRP context, however, each node $v \in \mathcal{V}$ represents a triplet $\Gamma_v, \gamma_v, q_{\Gamma_v}^{\gamma_v}$, where $\Gamma_v \in \mathcal{N}$ is a customer served during the last time period $\gamma_v \in \mathcal{T}$ with an amount $q_{\Gamma_v}^{\gamma_v}$. The arc set is defined as $\mathcal{A} = \{(a, b) \mid a, b \in \mathcal{V}', a \neq b \text{ and } \gamma_a \leq \gamma_b \text{ if } a, b \in \mathcal{V}\}$. Γ denotes a multi-period giant tour where each customer may appear more than once, but at most the number of available periods.

Determining when and how many products should be delivered to meet all demands while adhering to inventory capacities is necessary. Each customer can be supplied at most once per time period (no split delivery allowed) but may receive multiple deliveries over the time horizon, limited by the number of periods. The method involves (i) calculating delivery quantities as late as possible to avoid stockouts without considering the maximum inventory level and (ii) adjusting quantities that exceed the maximum inventory level as follows.

(i) It iterates through the time horizon from beginning to end and calculates, for each customer $i \in \mathcal{N}$, the specific time period $t \in \mathcal{T}$ when the inventory level for that customer would become negative based on its initial inventory and demands, assuming no deliveries are made. Consequently, the quantity q_i^t is determined for each customer $i \in \mathcal{N}$ during period $t \in \mathcal{T}$.

For each customer $i \in \mathcal{N}$ and time period $t \in \mathcal{T}$, if the inventory level I_i^{t-1} is lower than the demand d_i^t, a delivery is necessary. The total amount of goods delivered is based on the batch size, with the delivery amount being the smallest multiple of ℓ_i required to meet the demand, as specified in Equation (4.1).

$$q_i^t = \left\lceil \frac{d_i^t - I_i^{t-1}}{\ell_i} \right\rceil \ell_i \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{4.1}$$

Next, the inventory level at the end of period t is computed as the inventory level at the end of the previous period $t - 1$, plus the deliveries made during period t (q_i^t), minus the customer demand in period t (d_i^t). This calculation is represented by Equation (4.2) and is analogous to Constraint (2.25), excluding the vehicle assignment considerations. Algorithm 3 presents the steps of these calculations.

$$I_i^t = I_i^{t-1} + q_i^t - d_i^t \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{4.2}$$

Algorithm 3: Right-shift quantities

Data: *data*: problem data from Chapter 2
Result: q_i^t: amount to be delivered for customer $i \in \mathcal{N}$ in period $t \in \mathcal{T}$
I_i^t: inventory level for customer $i \in \mathcal{N}$ in period $t \in \mathcal{T}$

Begin
for each customer $i = 1$ *to* $|\mathcal{N}|$ **do**
  $I_i^0 \leftarrow s_i$
end
for each period $t = 1$ *to* $|\mathcal{T}|$ **do**
  for each customer $i = 1$ *to* $|\mathcal{N}|$ **do**
    $q_i^t \leftarrow 0$
    if $d_i^t > I_i^{t-1}$ **then**
      $q_i^t \leftarrow \left\lceil \frac{d_i^t - I_i^{t-1}}{\ell_i} \right\rceil \times \ell_i$
    end
    $I_i^t \leftarrow I_i^{t-1} + q_i^t - d_i^t$
  end
end
return $q_i^t, I_i^t \ \forall i \in \mathcal{N}, t \in \mathcal{T}$
End

(*ii*) Since the values q_i^t may exceed the maximum inventory level allowed per customer, a left-shifting of quantities from the end of the time horizon to the beginning may be necessary. For each customer $i \in \mathcal{N}$ and each period $t \in \mathcal{T}$, if $U_i < I_i^t$ (i.e., the inventory level surpasses the maximum allowed for the customer), then deliveries must be shifted. The amount to be shifted is determined by considering the customer's batch size and is adjusted to the minimum possible value.

Algorithm 4: Left-shift quantities

Data: *data*: problem data from Chapter 2
q_i^t: quantity for customer $i \in \mathcal{N}$ in period $t \in \mathcal{T}$ from Algorithm 3
I_i^t: inventory level for customer $i \in \mathcal{N}$ in period $t \in \mathcal{T}$ from Algorithm 3
Result: q_i^t: updated quantities

Begin
for each customer $i = 1$ *to* $|\mathcal{N}|$ **do**
  for each period $t = \mathcal{T}$ *to* 2 **do**
    $\delta_i^t \leftarrow U_i - I_i^t$
    if $\delta_i^t < 0$ **then**
      $p \leftarrow \left\lceil \frac{|\delta_i^t|}{\ell_i} \right\rceil \times \ell_i$
      $q_i^{t-1} \leftarrow q_i^{t-1} + p$
      $I_i^{t-1} \leftarrow I_i^{t-1} + p$
      $q_i^t \leftarrow q_i^t - p$
      $I_i^t \leftarrow I_i^t - p$
    end
  end
end
return $q_i^t \ \forall i \in \mathcal{N}, t \in \mathcal{T}$
End

Secondly, once the delivery quantities are established, the giant tour $\Gamma = \Gamma^1, \ldots, \Gamma^t$ can be defined, where Γ^t represents a sequence of customers for a period $t \in \mathcal{T}$. Note that Γ^t

can be empty, indicating that there are no customers to deliver in period t because their inventory is sufficient. Furthermore, Γ^t may vary from period to period, as stockouts do not occur simultaneously for all customers due to differences in their initial inventory levels and demands.

Based on the previously calculated quantities q_i^t, for each $i \in \mathcal{N}$ and $t \in \mathcal{T}$, we define $\mathcal{C}^t = i \in \mathcal{N} \mid q_i^t \neq 0$ as the set of customers to be delivered in period t, with $n^t = |\mathcal{C}^t|$. Similar to Γ^t, the set $\mathcal{C}^t$ may also be empty for the same reason. We denote Γ_n^t as the customer scheduled at period t in position n of Γ, where $n \in 0, |\Gamma| + 1$, and $|\Gamma| = \sum_{t \in \mathcal{T}} n^t + 1$, with an additional fictional node 0 at the beginning.

The giant tour Γ is constructed from the sets $\mathcal{C}^t$ in chronological order. For the first period t in which at least one delivery occurs, the customers in $\mathcal{C}^t$ are randomly ordered in Γ. For subsequent periods, Γ is arranged to facilitate the future Split algorithm evaluation, ensuring that routes start with a customer scheduled in period t and end with another customer assigned to a later period t' with $t' > t$.

Incorporating a customer from period t' into a route scheduled for period t incurs additional transportation costs if the customer has not yet been visited on this route. If the customer is already on the route, adding them only increases the delivery quantity without incurring extra transportation costs. This placement must also comply with other constraints, such as maximum inventory levels and vehicle capacity.

Thus, for a period $t' \in \mathcal{T}$ with $t' > t$ and for customers in $\mathcal{C}^t \cap \mathcal{C}^{t'}$, each customer is placed in the first available position in Γ with a probability λ. For each period from t' to T, customers from $\mathcal{C}^t \cap \mathcal{C}^{t'}$ are considered and placed to minimize costs. These customers are positioned in the first $|\mathcal{C}^t \cap \mathcal{C}^{t'}|$ positions according to probability λ. If placement fails, the next available slot is tested, and if all slots are filled, the process restarts from the first available position. Note that each customer in $\mathcal{C}^t \cap \mathcal{C}^{t'}$ is placed as close as possible to their appearance in the previous set $\mathcal{C}^t$, aligning with the scenario depicted in Figure 4.3.

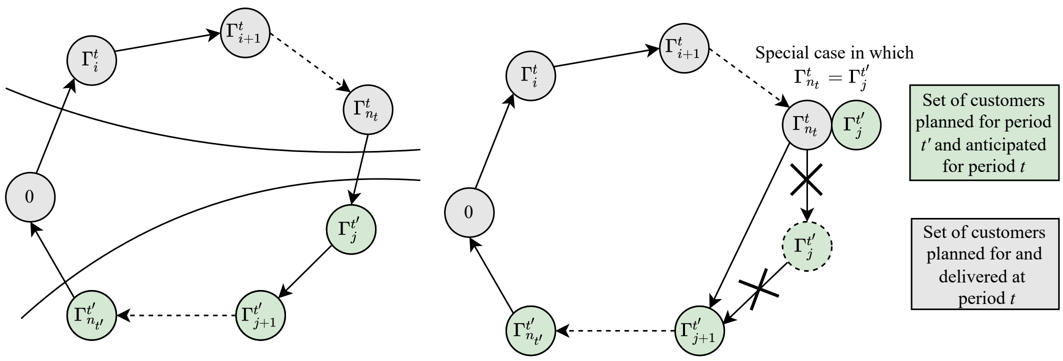


Figure 4.3: Gain in routing cost when applying a λ probability

Figure 4.3 depicts a scenario where strategically placing customers enhances the probability of having two deliveries for the same customer in consecutive periods appear next to each other in the giant tour. This arrangement can help reduce transportation costs when the Split algorithm is applied (as discussed in Section 4.3.2).

Once all customers from $\mathcal{C}^t \cap \mathcal{C}^{t'}$ have been positioned, the remaining customers in $\mathcal{C}^{t'} - (\mathcal{C}^t \cap \mathcal{C}^{t'})$ are inserted into random positions within Γ with equal probability. This placement process is repeated as needed until every period in $\mathcal{T}$ is addressed. Figure 4.4 demonstrates this approach with 6 customers, A, B, C, D, E, and F, across 4 time periods.

In Figure 4.4, the composition of sets $\mathcal{C}^t$ is shown on the left. Note that $\mathcal{C}^2$ is empty and

$$\begin{aligned}
\mathcal{C}^1 &= \{A, B, D\} \\
\mathcal{C}^2 &= \varnothing \\
\mathcal{C}^3 &= \{A, C, D, E, F\} \\
\mathcal{C}^4 &= \{A, B, C, D, F\}
\end{aligned}$$

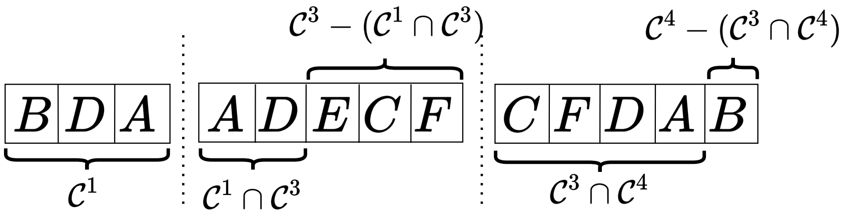


Figure 4.4: Customers placement into Γ

the final multi-period giant tour contains only period 1, 3 and 4. For period 1, customers A, B and D are randomly placed and the sequence results in BDA. In period 3, the 5 customers scheduled are organized as follows: first, customers from $\mathcal{C}^1 \cap \mathcal{C}^3 = A, D$ are inserted according to a probability λ to favor their placement closer to their occurrence in period 1 and it results in AD; secondly, the remaining customers $\mathcal{C}^3 - (\mathcal{C}^1 \cap \mathcal{C}^3) = C, E, F$ are randomly placed in the remaining cases. Lastly, in period 4, the same occurs: customers from $\mathcal{C}^3 \cap \mathcal{C}^4 = A, C, D, F$ are placed according to a λ probability and the only remaining $\mathcal{C}^4 - (\mathcal{C}^3 \cap \mathcal{C}^4) = B$ is inserted into the last available giant tour position.

Note that the fact of not having customers scheduled for period 2 in the giant tour does not exclude the possibility of having it once a feasible solution is obtained. That will be further explained in the nest Subsection concerning the giant tour partitioning into routes and may occur if it provides a less costly solution.

Algorithm 5: Customers sequencing in the giant tour

Data: *data*: problem data from Chapter 2
Result: q_i^t: amount to be delivered for customer $i \in \mathcal{N}$ in period $t \in \mathcal{T}$
I_i^t: inventory level for customer $i \in \mathcal{N}$ in period $t \in \mathcal{T}$

Begin
`/* Part 1. Random customers placement */`
$t \leftarrow$ get first period in which $|C^t| \neq \varnothing$
while $C^t \neq \varnothing$ **do**
  $c \leftarrow$ get a random customer in C^t
  Add the triplet $\{c, q_c^t, t\}$ at the next position p in Γ of $\{\Gamma_p, q_p, \gamma_p\}$, $p \in [1, |C^t|]$
  Remove customer c from C^t
end
`/* Part 2. "Mirror" random placement */`
$t' \leftarrow$ get next period in which $|C^{t'}| \neq \varnothing$
$p \leftarrow$ next available position of Γ (i.e., $|C^t| + 1$)
while the current period $t' \neq T$ **do**
  `/* Part 2.1. According to a probability` λ `*/`
  Define set $E = C^t \cap C^{t'}$
  Define set $F = C^{t'} - E$
  while $E \neq \varnothing$ **do**
    $e \leftarrow$ get the closest customer in E from its appearance in C^t
    repeat
      with a probability λ and if Γ_p is empty **do**
        Add the triplet $\{e, q_e^{t'}, t'\}$ at position p of $\{\Gamma_p, q_p, \gamma_p\}$
        Remove e from E
      with a probability $1 - \lambda$ **do**
        Increment the value of p
      if p value exceeds $[|C^t| + 1, |C^t| + |C^{t'}|]$ **then**
        Reset p to $|C^t| + 1$
      end
    until e is placed successfully;
  end
  `/* Part 2.2. Random probability */`
  $p \leftarrow$ next available position of Γ (i.e., $\sum_{t=1}^{t'} |C^t| + 1$)
  while $F \neq \varnothing$ **do**
    $f \leftarrow$ get a random customer in F
    Add the triplet $\{f, q_f^{t'}, t'\}$ at the next available position p of $\{\Gamma_p, q_p, \gamma_p\}$, $p \in [|C^{t'}|, |C^{t'}| + |C^t|]$
    Remove customer f from F
  end
  $t \leftarrow t'$
  $t' \leftarrow$ get next period in which $|C^t| \neq \varnothing$
end
return Γ, γ_i *for all* $i \in \{1, \ldots, |\Gamma|\}$
End

4.3.2 Splitting the giant tour

Given the auxiliary directed graph $G = (\mathcal{V}', \mathcal{A})$, an arc (u, v) represents a sub-sequence $\sigma = (\Gamma_{u+1}, \ldots, \Gamma_v)$ of consecutive customers visited within a single route assigned to period γ_{u+1}, and includes $\sum_{u'=u+1}^{v} q_{\Gamma_{u'}}^{\gamma_{u'}}$ products. A valid solution for the HIRP-BS can be derived from the giant tour Γ. Throughout the search process, a list of labels is maintained at each node of Γ, facilitating the management of resources and inventory. Each label contains information regarding the previous and current routes originating from its node and includes:

- t: the assignment period equivalent to the period of the route origin node;
- I_i^t: the inventory level for each customer and supplier at period t;
- $cost$: the accumulated routing and inventory costs;
- k: the type of vehicle used at period t;
- $M^{k,t}$: the number of vehicles of type k available at period t;
- sc_i^t: binary indicator equal to 1 if customer i is serviced at period t and 0 otherwise;
- fn, idx: the origin node of the route and its index, respectively.

Since each label in a partially feasible solution corresponds to a route, it holds information about the inventory levels for each customer and the time period of the route. This is necessary because the labels at a given node can span different periods. As a result, varying values for accumulated inventory levels can appear. This variability occurs because the partitioning algorithm may consider the possibility of advancing customer deliveries, leading to a situation where labels at the same node reflect different periods and inventory levels.

Thus, an optimal solution for the HIRP-BS, based on the giant tour Γ, corresponds to a minimum-cost path that incorporates both resource (vehicle) usage and inventory management (supplier/customer inventory constraints) and with no maximum number of labels allowed per node of Γ. It is important to note that while constructing a feasible solution for any configuration of Γ, skipping any nodes to form a route is not permitted. This highlights the significant influence of the giant tour on the quality of a feasible solution. □

Generating a label

Creating a label l from a node $i \in \mathcal{V}$ to $j \in \mathcal{V}$ involves adding a label at node $j \in \mathcal{V}$ label list $\mathcal{L}_j$ based on a label from node i, using a vehicle of type $k \in \mathcal{K}^t$ that is available during period $t \in \mathcal{T}$. This process assumes that the propagation from i to $j-1$ has already been completed and can be viewed as an update from each label incident to a node sequentially.

Two scenarios need to be considered. The first scenario occurs when both customers Γ_{i+1} and Γ_j, the start and end of the arc (i, j), are assigned to the same time period. This is the common case, analogous to the HVRP algorithm. The second scenario arises when Γ_{i+1} and Γ_j are assigned to different time periods, which is a specific aspect of the IRP and a central focus of the algorithm presented in this thesis. This particular case will be explored further in this chapter.

In this context, a route from customer Γ_{i+1} to Γ_j incurs routing and inventory costs as detailed below:

The routing cost, composed of the fixed cost $f^{k,t}$ of the vehicle k at period t, added to the variable transportation cost that depends on the vehicle used:

$$f^{k,t} + c_{0,\Gamma_{i+1}}\ v^{k,t} + \sum_{u=i+1}^{j} c_{\Gamma_u,\Gamma_{u+1}}\ v^{k,\gamma_u} + c_{\Gamma_j,0}\ v^{k,\gamma_u} \tag{4.3}$$

The inventory cost contains two parts: the increase in the inventory cost for each customer $\Gamma_u, u \in \{i+1, ..., j\}$ of a given route as a new quantity $q^{\gamma_u}_{\Gamma_u}$ is delivered and the one that represents the decrease in the inventory cost at the supplier since the quantity delivered to the customer was taken from its inventory:

$$\sum_{t=\Gamma_{i+1}}^{T} \sum_{u=i+1}^{j} q^{\gamma_u}_{\Gamma_u}\ h^{\gamma_u}_{\Gamma_u} - \sum_{t=t'}^{T} \sum_{u=i+1}^{j} q^{\gamma_u}_{\Gamma_u}\ h^{\gamma_u}_{0} \tag{4.4}$$

Once the costs are established, the inventory level is calculated as follows:

$$I^{\gamma_u}_{\Gamma_u} = I^{\gamma_u-1}_{\Gamma_u} + q^{\gamma_u}_{\Gamma_u} - d^{\gamma_u}_{\Gamma_u} \quad \forall u \in \{i+1, ..., j\} \tag{4.5}$$

□

Generation of first label (node 0)

The first and only label to node 0 of the giant tour is defined as:

$$(t, I^t_i, cost, v, M^{k,t}, sc^t_i, fn, idx) = (-, I^t_i, -, M^{k,t}, -, I^t_i, 0, 0) \tag{4.6}$$

In this case, neither period t nor vehicle v is assigned, as it does not pertain to any route. The inventory level is calculated by considering the initial inventory, customer demands, and supplier production capacity according to the procedure outlined in Equations 4.7. The *cost* is determined by Equations 4.3 and 4.4, taking into account the previously defined inventory level I^t_i. The available vehicles $M^{k,t}$ are those specified by the problem data (see Section 2.3 in Chapter 2). The binary indicator is set to zero ($sc^t_i = 0\ \forall i \in \mathcal{N}, t \in \mathcal{T}$), and both the father and index node are set to 0 since there is no preceding node.

Equations 4.7 allows the definition of the inventory level in a scenario where no delivery operation is performed. The first equation defines the initial inventory level given by the problem data and expressed by $s_i\ \forall i \in \mathcal{N}'$ (both customers and supplier). The second concerns only the customers and adds the supplier production capacity per period which is also a problem data and given by $r^t\ \forall t \in \mathcal{T}$. The third and last one refers only to the customers and take into account the demands over the time periods given by the problem data and expressed by d^t_i.

$$I^t_i = \begin{cases} 1^{st}: \ I^0_i = s_i & \forall i \in \mathcal{N}' \\ 2^{nd}: \ I^t_0 = I^{t-1}_0 + r^t & \forall t \in \mathcal{T} \\ 3^{rd}: \ I^t_i = I^{t-1}_i - d^t_i & \forall i \in \mathcal{N}, t \in \mathcal{T} \end{cases} \tag{4.7}$$

Note that, according to Equations 4.7, customer inventory levels may be negative, as no delivery operations are considered. The purpose of this step is to perform pre-calculations that help with computations during label propagation in the Split algorithm and to avoid redundant recalculations.

□

Addition of a label

After augmenting the new candidate label with additional information, its feasibility is tested by checking that the inventory levels remain within the specified lower and upper bounds and comply with the supplier production capacity. The feasibility check also includes verifying vehicle capacities and availability, and ensuring that split delivery constraints are met, which prevent a customer from being visited on multiple routes within the same period. Once the feasibility is confirmed, the candidate label is added to the node list. It is important to note that batch size constraints do not need further verification, as the delivery quantities in the giant tour were previously calculated to adhere to these constraints.

To avoid excessive computational time during label propagation, the number of labels associated with any node in the giant tour is restricted by the parameter $nMaxLabel$. Labels are sorted by ascending cost to prioritize the exploration of lower-cost labels. Consequently, when a label is added, its position in the list is determined by its cost—lower costs correspond to higher priority positions in the list.

□

Label propagation

The Split algorithm propagates labels to construct a feasible solution for the HIRP-BS. The algorithm primarily involves four nested loops (see Algorithm 6), iterating over graph nodes ($u \in \mathcal{V}$), customers ($\Gamma_u \in \mathcal{N}$), labels available at each node list ($j \in \mathcal{V}$ and $\mathcal{L}_j$), and vehicle types ($k \in \mathcal{K}^t$) in that order. The overall process is depicted in Figure 4.5.

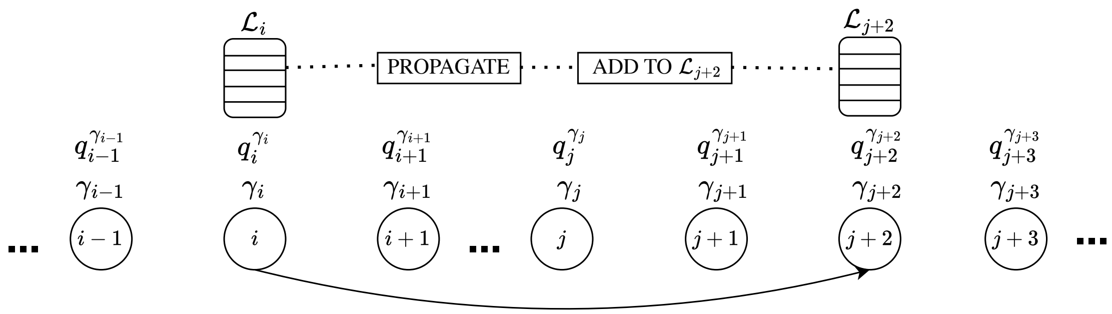


Figure 4.5: Labels propagation

Figure 4.5 illustrates the propagation of labels from node i to $j+2$. In this scenario, a label is selected from the list associated with node i, denoted $\mathcal{L}i$. This label is updated based on the customers in the candidate route, which includes nodes i, $i+1$, ..., $j+2$, as well as the scheduled quantities $q_i^{\gamma_i}$, $qi+1^{\gamma_{i+1}}$, ..., $q_{j+2}^{\gamma_{j+2}}$, and the periods γ_i, γ_{i+1}, ..., γ_{j+2}. If the updated label is feasible, it is then added to the list $\mathcal{L}_{j+2}$.

Algorithm 6: Split algorithm

Data: data from the problem, triplets $\{\Gamma, \gamma, q\}$, maximum number of labels $\mathcal{L}^{max}$
Result: a feasible solution s

```
Begin
Generate the only label at L_0
for each node v ∈ |V| − 1 of Γ do
    Get customer j = Γ_{v+1} and period t = γ_{v+1}
    Set inventory and routing costs and volume load to zero
    repeat
        Calculate inventory costs and distance traveled
        Update vehicle load with q_j^t
        for each label l ∈ L_j do
            Define a label L = l
            Update L
            for each vehicle v ∈ K^t do
                Update the inventory and routing costs
                if L is feasible then
                    Add label to L_j considering L^max
                end
            end
        end
    until Δ^load > argmax_{k∈K^t}{B^k} and j ≥ |Γ|;
end
Retrieve critical path and build the solution s
return s if Γ_{|L_i|} > 0 and ∅ otherwise
End
```

The instructions from Algorithm 6 are detailed below.

- **Generate the only label at $\mathcal{L}_0$.** List $\mathcal{L}_0$ has no more than one unique label. It happens because no propagation has been done so far. The details of this step are detailed in Subsection 4.3.2 (Generation of first label (node 0))

- **Set inventory and routing costs and volume load to zero.** The propagation restarts from a node v, which means a new route is about to be created. For that reason, the inventory costs, the routing costs and the volume load are set to zero to save later the information about the new route.

- **Calculate inventory costs and distance traveled.** Since a node representing a customer is added into a new (if $j = v + 1$) or existing route (if $j > v + 1$), the involved inventory costs are calculated according to Equations 4.4. Special cases in which the origin node and the its successor are (Case 1) or not (Case 2) in the same period are further detailed by Equations 4.9 for the inventory costs. At this stage, the routing costs can only be considered according to the distance traveled. Neither the fixed nor the variables costs can be calculated for the moment since no vehicle has been assigned to the route under definition.

- **Update vehicle load with q_j^t.** As for the costs, the vehicle load set to zero once a new route is created is incremented according to the quantity of products from node Γ_j. Note that in this case, the load does not depend if it is the first customer or not on the route.

- **Define a label** L**.** Considering all the labels incident to L_v, each one is treated separately and copied to L to be updated.
- **Update** L**.** The copy label L that has just been created is updated according to the previous information (inventory and routing costs and the vehicle load), as well as the assignment period (t), the type of vehicle used (v), the number of vehicles available ($M^{k,t}$), the indicators of visitation (sc_i^t), origin node (fn) and its index (idx). Also, according to Equation 4.5, the inventory levels are also updated.
- **Update the inventory and routing costs.** The inventory costs are added according to the precedent values and the routing costs can now be defined according to the vehicle that is being tested to be assigned to the route. The corresponding routing costs are then expressed by Equation 4.8. Special cases rely on the fact that if the customer already exists on the route, the routing cost does not change since the visitation has already been considered before.
- **If feasible, add label to** $\mathcal{L}_i$**.** Once the label is created, it is time to check its feasibility. The following tests are performed:
  - The inventory level must be under or equal to the maximum level authorized and always bigger than zero for the customers. For the supplier, it should be under its inventory level considering the production capacity.
  - The availability of vehicles. For all nodes except the last one in which no more routes can be created, a label is unfeasible if there are no more vehicles available of any type considered.
  - If the customer has already been visited in another route at the same period, the label is unfeasible since no split delivery is allowed. The visitation indicator sc_i^t provides this information.
- **Retrieve critical path and build the solution** s**.** Once the search has been finished, the critical path can be retrieved if and only if all the nodes of the giant tour have been explored. Note that a giant tour may lead to an unfeasible solution if at least the last node does not contain any label. Otherwise, a solution can be defined using the first label incident to the last node of the giant tour. The path can be retrieved with the information from the parent node (fn) and its index (idx) label. In other words, the critical path corresponds to the shortest path found considering a given giant tour.

Further details on the special cases while exploring the giant tour nodes and the information they contain are provided below.

□

Once iterating over the graph nodes, if an arc (v, u) satisfies $u = v + 1$, this implies that the route consists of a single customer, Γ_u, with a vehicle type k assigned. The routing cost for this specific case can then be computed by considering both fixed and variable costs, as detailed in Equation 4.8. This represents a specific instance of the general case provided in Equation 4.3.

$$routing_{cost} = f^{k,t} + c_{0,\Gamma_u}\ v^{k,t} + c_{\Gamma_u,0}\ v^{k,t} \tag{4.8}$$

The inventory cost ($inventory_{cost}$) can thus be defined by Equation 4.9.

$$inventory_{cost} = \sum_{t'=t}^{T} q_{\Gamma_u}^{t'}\ h_{\Gamma_u}^{t'} - \sum_{t'=t}^{T} q_{\Gamma_u}^{t'}\ h_0^{t'} \tag{4.9}$$

The supplier inventory level is then updated according to Equation 4.10 if the only customer Γ_u is the first visited in period t' (i.e., $\sum_{i\in\mathcal{N}} sc_i^{t'} = 0$). Otherwise, (i.e., $\sum_{i\in\mathcal{N}} sc_i^{t'} > 0$), then the supplier production capacity has already been considered, which means that Equation 4.11 is used.

$$I_0^{t'} = I_0^{t'-1} - q_{\Gamma_u}^{\gamma_u} + r^{\gamma_u} \tag{4.10}$$

$$I_0^{t'} = I_0^{t'-1} - q_{\Gamma_u}^{\gamma_u} \tag{4.11}$$

The accumulated amount of products must be stored in order to test later if the vehicle capacity constraints are respected. Thus, for a given label L, its attributes can be calculated as:

$$L.t = \gamma_u \tag{4.12}$$

$$L.cost = P.cost + routing_{cost} + inventory_{cost} \tag{4.13}$$

$$L.I_{\Gamma_u}^{\gamma_u} = P.I_{\Gamma_u}^{\gamma_u} + q_{\Gamma_u}^{\gamma_u} - d_{\Gamma_u}^{\gamma_u} \tag{4.14}$$

A customer Γ_u is then added to a route scheduled at period γ_i and sc_i^t is set to 1 which is the last period t where the customer has been serviced: by consequence, the demands $d_{\Gamma_u}^t$ must be subtracted for all the periods from γ_i period and the period where $sc_i^t = 1$.

$$L.I_{\Gamma_i}^{\gamma_i} = P.I_{\Gamma_i}^{\gamma_i} + q_{\Gamma_i}^{\gamma_i} - \sum_{t=t'|L.sc_i^t=1}^{t=\gamma_i} d_{\Gamma_i}^{\gamma_i} \tag{4.15}$$

Note that $P.I_{\Gamma_i}^{\gamma_i}$ is equivalent to $P.I_{\Gamma_i}^{\gamma_i-1}$ since the customer appear only once per period and $P.I_{\Gamma_i}^{\gamma_i}$ has not been updated before in the current period.

$$L.I_0^{\gamma_i} = P.I_0^{\gamma_i} - q_{\Gamma_u} \tag{4.16}$$

$$L.sc_{\Gamma_u}^t = 1 \tag{4.17}$$

$$L.M^{k,t} = P.M^{k,t} - 1 \tag{4.18}$$

If an arc (v, u) satisfies $u > v + 1$, this indicates that the route is extended by incorporating a new customer, Γ_u, after completing the evaluation of the route that starts at Γ_{v+1} and ends at Γ_{u-1}.

Since the Split algorithm explores label propagation from node i to j and then from node j to $j + 1$, the computation of all labels can be performed in $\mathcal{O}(1)$ time.

At this point, there are two possible cases:

Case 1. If Γ_j is in the same period t as Γ_i (where t is the period of the current route), incorporating Γ_j into the route necessitates updating the label as follows:

To calculate the routing cost, subtract the cost of the previous routes that end at node $j+2$ (customer Γ_{j+2}) from the depot node. Then, add the cost of two new arcs: one from customer Γ_{j+2} to customer Γ_{j+3}, and another from customer Γ_{j+3} to the depot.

$$routing_{cost} = routing_{cost} + (-c_{j+2,0}\ v^{k,t'} + c_{\Gamma_{j+2},\Gamma_{j+3}}\ v^{k,t'} + c_{\Gamma_{j+3},0}\ v^{k,t'}) \tag{4.19}$$

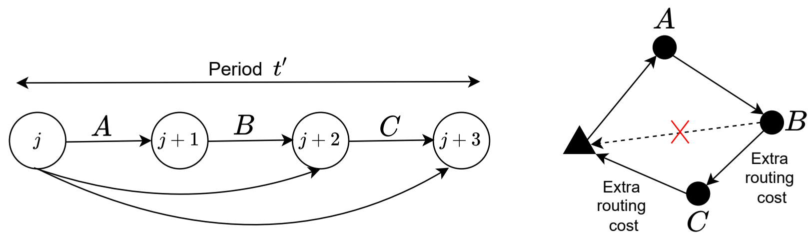


Figure 4.6: Addition of customer Γ_{j+3} in the route

The inventory cost is computed considering the $inventory_{cost}$ of the route that ends a node $j-1$ plus the new inventory holding cost at customer Γ_j minus the inventory holding cost at the supplier, which decreases of $\sum_{t=t'}^{T} q_{\Gamma_u}^{t'}\ h_0^{t'}$.

$$inventory_{cost} = inventory_{cost} + \sum_{t=t'}^{T} q_{\Gamma_u}^{t}\ h_{\Gamma_u}^{t} - \sum_{t=t'}^{T} q_{\Gamma_u}^{t}\ h_0^{t} \tag{4.20}$$

The inventory levels at both customer (Γ_j) and supplier are given by the increase and the decrease of $q_{\Gamma_j}^{t}$ unities, respectively.

$$I_{\Gamma_u}^{t'} = I_{\Gamma_u}^{t'} + q_{\Gamma_u}^{t'} - d_{\Gamma_u}^{t'} \quad \forall u = i+1, ..., j-1 \tag{4.21}$$

$$I_0^{t'} = I_0^{t'} - \sum_{u=i+1}^{j} q_{\Gamma_u}^{t'} \tag{4.22}$$

Thus, the new label modeling the route that starts at node i, ends at node j and propagates the label L (coming from node i) is defined as follows:

$$L.t = \gamma_i \tag{4.23}$$

$$L.cost = P.cost+, routing_{cost} + inventory_{cost} \tag{4.24}$$

$$L.I_{\Gamma_u}^{t'} = P.I_{\Gamma_u}^{t'} - \sum_{t=t'|L.sc_i^{t'}=1}^{t=\gamma_i} d_{\Gamma_u}^{t} \forall u = i+1, ..., j-1 \tag{4.25}$$

$$L.I_0^{\gamma_i} = P.I_0^{\gamma_i} - \sum_{u=i+1}^{t} q_{\Gamma_u}^{\gamma_i} \tag{4.26}$$

$$L.sc_i^{t} = 1 \tag{4.27}$$

$$L.M^{k,t} = P.M^{k,t} - 1 \tag{4.28}$$

□

Case 2. Γ_i and Γ_u are not in the same period. Two different sub cases must me considered, as follows.

Case 2.1. $\Gamma_j \neq \Gamma_i$ with $u = i+1, \ldots, j$ (the new customer Γ_u does not appear in the route).

Such a situation is similar to Case 1; the updated label is the same.

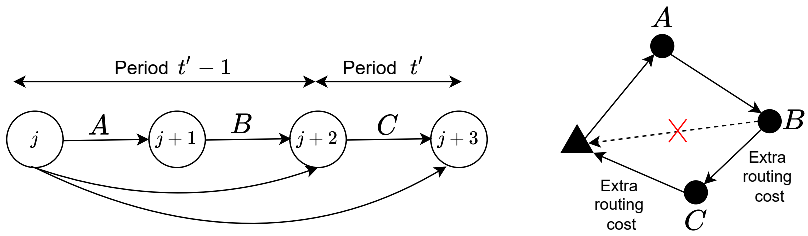


Figure 4.7: Addition of customer Γ_{j+3} in the route

Case 2.2. $\Gamma_j = \Gamma_v$ with $v = j+1, \ldots, j+3$.

Such a situation means that the customer $j+3$ is already present in the route that starts at node j and ends at node $j+2$. Therefore, the routing cost does not change.

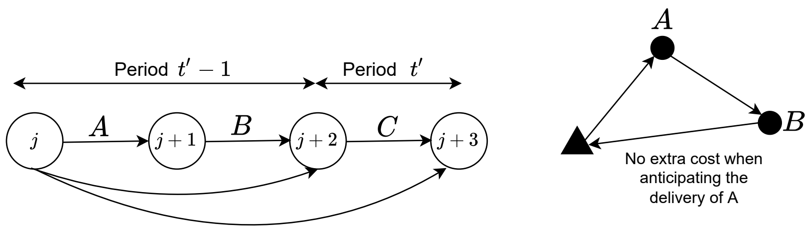


Figure 4.8: Delivery anticipation when adding an already visited customer

The inventory cost is computed considering the $inventory_{cost}$ of the trip that ends at node $j-1$ plus the new inventory holding cost at customer Γ_j minus the inventory holding cost at the supplier, which decreases of $\sum_{t=t'}^{T} q_{\Gamma_u}^{t'} \; h_0^{t'}$.

$$inventory_{cost} = inventory_{cost} + \sum_{t=t'}^{T} q_{\Gamma_u}^{t} \; h_{\Gamma_u}^{t} - \sum_{t=t'}^{T} q_{\Gamma_u}^{t} \; h_0^{t} \tag{4.29}$$

Similarly to the precedent case, the inventory level at both customer Γ_j and supplier Γ_0 increase and decrease of $q_{\Gamma_j}^{t}$ unities, respectively.

$$I_{\Gamma_u}^{t'} = I_{\Gamma_u}^{t'} + q_{\Gamma_u}^{t'} - d_{\Gamma_u}^{t'} \quad \forall u = i+1, ..., j-1 \tag{4.30}$$

$$I_0^{t'} = I_0^{t'} - \sum_{u=i+1}^{j} q_{\Gamma_u}^{t'} \tag{4.31}$$

The generation of the new label is achieved as follows:

$$L.t = \gamma_i \tag{4.32}$$

$$L.cost = P.cost + routing_{cost} + inventory_{cost} \tag{4.33}$$

$$L.I_{\Gamma_u}^{t'} = P.I_{\Gamma_u}^{t'} - \sum_{t=t'|L.sc_i^{t'}=1}^{t=\gamma_i} d_{\Gamma_u}^{t} \forall u = i+1, ..., j-1 \tag{4.34}$$

In this specific situation, $\sum_{t=t'|L.sc_i^{t'}=1}^{t=\gamma_i} d_{\Gamma_u}^{t} = 0$ because $L.sc_i^{t'} = 1$ for $t' = \gamma_i$. Consequently:

$$L.I_0^{\gamma_i} = P.I_0^{\gamma_i} - \sum_{u=i+1}^{t} q_{\Gamma_u}^{\gamma_i} \tag{4.35}$$

$$L.sc_i^t = 1 \tag{4.36}$$

$$L.M^{k,t} = P.M^{k,t} - 1 \tag{4.37}$$

In routing problems such as the CVRP and HVRP, the Split algorithm leverages dominance rules to eliminate certain labels. These rules help to prune labels that are either dominated by others or are less optimal compared to existing labels, based on the number of available vehicles and the partial solution cost associated with each label. However, for Inventory Routing Problems (IRPs), such dominance rules are not applicable. This is because the information related to inventory levels can fluctuate throughout the giant tour exploration. Therefore, it is challenging to determine if a particular inventory level is advantageous or not for pruning purposes. In routing problems, resources are defined by the number of vehicles available, which only decreases as new labels are added. In contrast, in IRPs, inventory levels can increase when additional quantities are added to a customer or decrease when demands are subtracted, making it difficult to establish clear dominance rules.

At each node, labels are sorted in ascending order based on their total cost, which includes both inventory and routing costs. When the list of labels for a node reaches its maximum size and a new candidate label with a lower cost than at least one existing label is found to be feasible, the label with the highest cost is discarded. While this approach may lead to a sub-optimal splitting of the giant tour, it is generally not a significant drawback. Combining this algorithm with additional techniques, such as local search methods, can still yield high-quality solutions, as demonstrated by Boudia et al. (2007).

4.3.3 Shortest path retrieval

After executing the Split algorithm, the solution can be reconstructed by tracing through the parents and indices of the labels. Begin with the first label associated with the last node of the giant tour, which is the one with the minimum cost due to the labels being sorted by ascending solution cost. From this starting point, you can retrieve the routes, the assigned periods, and the vehicle types used. Figure 4.9 demonstrates how to trace the path from the last node back to the first node in the giant tour to reconstruct the solution.

A solution s for the HIRP-BS contains a set of routes per period $t \in \mathcal{T}$ and each route is composed of a set of customers, an assigned vehicle type, and a total load of products to be delivered for each customer. In the giant tour illustrated by Figure 4.9, the first label (the minimum-cost label) in $\mathcal{L}_9$ has node 5 as parent and 3 as index since the third label in $\mathcal{L}_5$ originated the route $\{0, E, D, 0\}$ and is assigned to period 2 since the parent node belongs to this period. Then, route $\{0, B, C, A, 0\}$ is retrieved by the second label in $\mathcal{L}_1$ and belongs to period 1. The third and last route $\{0, C, 0\}$ has its origin in the fourth label in list $\mathcal{L}_0$ and is also assigned to period 1.

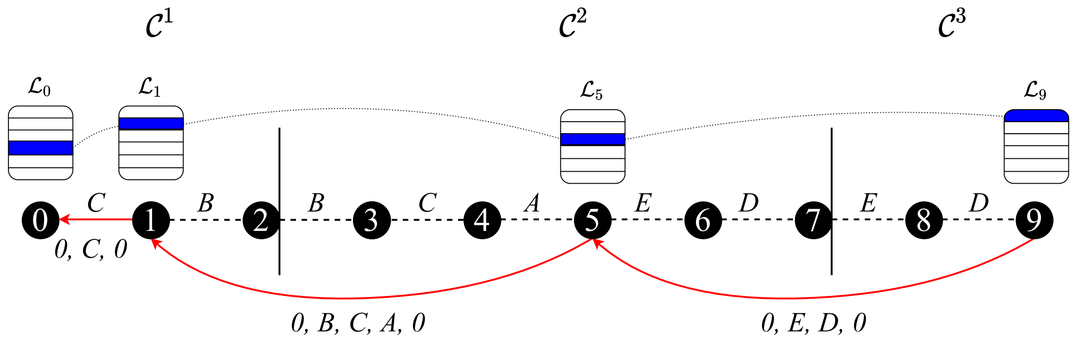


Figure 4.9: Retrieving the critical path

The Split algorithm may succeed or not in the labels exploration. If it is the case as in the previous example, the set of routes can be retrieved. If not, only partial routes are retrieved and, consequently, the solution is not feasible since demands were not met completely. It can arises due to the following non-exhaustive reasons:

- an insufficient number of labels stored at each node since keeping all of the possible labels would increase exponentially the space complexity of the algorithm and finding a feasible solution would be harder.
- the giant tour generated is infeasible because the order of the nodes should affect the splitting phase and may stop the search due to
  - an insufficient number of vehicles available at a given period **or**
  - the quantities previously defined that does not take into account the inventory management.

4.4 Evolutionary Local Search algorithm

The Evolutionary Local Search (ELS) consists of mutating and applying local search operators seeking for an improved solution. It was introduced by Wolf and Merz (2007) to treat two problems named the Super-Peer Selection and p-Hub Median. Essentially, the ELS is composed of two phases: mutation and local search. The objective is to explore the solutions neighborhood and avoid staying stuck in a basin (which can yield to local minima) in order to diversify the solutions.

Figure 4.10 illustrates how local search and mutation influence the search for a global optima. Local search operators allows exploring a solution neighborhood seeking for the best solution available in a basin whereas mutation can skip from one neighborhood to another one (from one basin to another) to increase diversification in the solutions and avoid local minima since the objective is to reach the global optima.

Algorithm 7 illustrates the overall idea of the ELS. As input parameters, the algorithm considers both the number of global ngi and local ni iterations and an initial solution s previously defined. Two nested loops are considered to iterative over the different ELS levels. Lines 4 to 12 refers to one ELS level and the best solution obtained at each is stored at $\bar{s}$ and $\bar{f}$ and corresponds to a local minima. At each execution of loop from lines 2 to 16, the cost of the best solution found is reseted (line 3) and it allows to select at the end the best solution from all the ELS levels expressed by s^*.

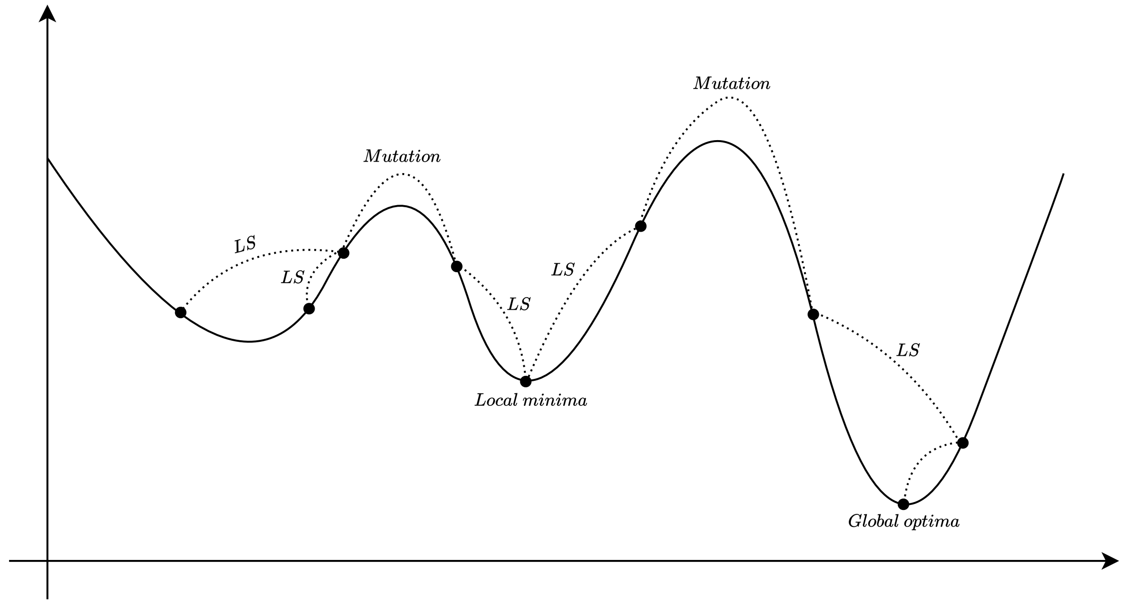


Figure 4.10: Walks on the solutions search space

Algorithm 7: Evolutionary Local Search

Data: ngi: number of global iterations, ni: number of iterations, s: initial solution
Result: an improved solution s^*

Begin
for each global iteration itg from 1 to ngi **do**
  $\bar{f} \leftarrow \infty$
  for each iteration it from 1 to ni **do**
    $s' \leftarrow s$
    Mutate(s')
    LocalSearch(s')
    if $f(s') < \bar{f}$ **then**
      $\bar{s} \leftarrow s'$
      $\bar{f} \leftarrow f(s')$
    end
  end
  if $\bar{f} < f(s^*)$ **then**
    $s^* \leftarrow \bar{s}$
  end
end
return s^*
End

For the IRP, and considering the algorithm Split previously presented in this Chapter, one intermediate phase is added between the mutation and the local search steps. It consists in the evaluation phase which corresponds to the use of the Split algorithm to recreate a solution. The three steps are described below.

The mutation phase corresponds to the permutation of $nMut$ customers in the giant tour and is performed within the customers from the same period to avoid stock disruption.

The evaluation phase comes to the use of the giant tour in the sequence previously mutated, and lastly, the local search consists of an adaptive schema that aims to improve the composition of routes previously found by the Split and is explained in the next section.

4.4.1 Mutation

The mutation corresponds to a swap operator between two nodes of the giant tour. Two nodes are selected and are then permuted. Note that one node corresponds to the triplet $\{\Gamma_v, \gamma_v, q_{\Gamma_v}^{\gamma_v}\}$. Considering the generation of the giant tour and in order to avoid extra calculations and inventory disruption, this operator is applied only among nodes within the same period of time. And to provide diversification, $nMut$ swaps are applied at mutation phase but not necessarily for the same period. Figure 4.11 illustrates the swap operator.

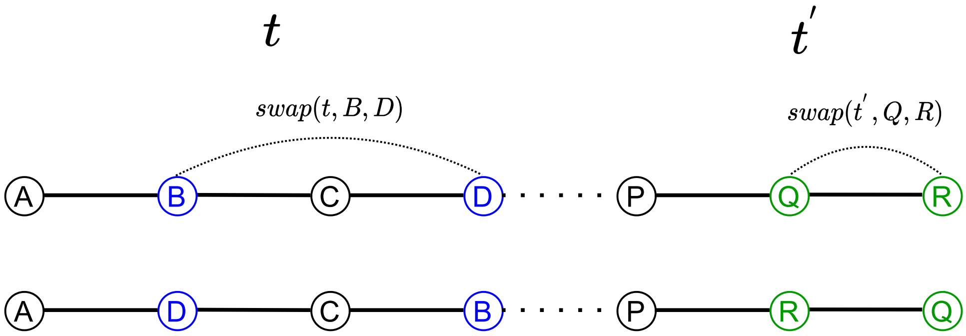


Figure 4.11: Swap mutation operator

In Figure 4.11, the operator is executed twice. Firstly between customers B and D assigned to period t and secondly for customers Q and R in period $t^{'}$. Algorithm 8 gives more details on how the mutation phase is executed.

Algorithm 8: Mutation

Data: $\Gamma = \{\Gamma_v, \gamma_v, q_{\Gamma_v}^{\gamma_v}\}$: the giant tour, $|\mathcal{T}|$: number of periods, $nMut$: the number of swap operations

Result: $\Gamma^{'}$: the giant tour modified

1 **Begin**
2 **for** each operation from 1 to $nMut$ **do**
3 $\quad \bar{t} \leftarrow random(1, |\mathcal{T}|)$
4 $\quad N_1, N_2 \leftarrow getTwoRandNodes(\bar{t}, \Gamma),\ N_1 \neq N_2$
5 $$swap(\bar{t}, N_1, N_2) \begin{cases} 1^{st} : \{\Gamma_{aux}, \gamma_{aux}, q_{\Gamma_{aux}}^{\gamma_{aux}}\} \leftarrow \{\Gamma_{N_1}, \bar{t}, q_{\Gamma_{N_1}}^{\bar{t}}\} \\ 2^{nd} : \{\Gamma_{N_1}, \bar{t}, q_{\Gamma_{N_1}}^{\bar{t}}\} \leftarrow \{\Gamma_{N_2}, \bar{t}, q_{\Gamma_{N_2}}^{\bar{t}}\} \\ 3^{rd} : \{\Gamma_{N_2}, \bar{t}, q_{\Gamma_{N_2}}^{\bar{t}}\} \leftarrow \{\Gamma_{aux}, \gamma_{aux}, q_{\Gamma_{aux}}^{\gamma_{aux}}\} \end{cases}$$
6 **end**
7 **return** $\Gamma^{'}$
8 **End**

As input, the algorithm needs the giant tour to be modified, the horizon of time as well as the number of swap operations to be executed. $nMut$ operators are applied (lines 2 to 6) and at each, a random period is chosen (line 3) and two customers N_1 and N_2 are chosen with $N_1 \neq N_2$ (line 4). Next, the swap operator is applied (line 5): at first, an auxiliary node stores the first node N_1 to be swapped; at second, node N_1 is replaced by N_2 and at last, node N_2 is replaced by the auxiliary which contains the N_1.

4.4.2 Evaluation

It consists in applying the Split algorithm presented in Subsection 4.3.2 in the giant tour that has been previously changed in the mutation phase in order to define a new solution based on the changes made by the successive swap operators. Note that the new solution obtained may be of a worst quality when compared to the one before the mutation. It occurs because when permuting successive nodes in the sequence, splitting it can lead to an unfeasible solution due to the insufficient number of labels per node as well as the availability of the vehicles and the customers inventory management as previously discussed in Subsection 4.3.3.

4.4.3 Local search

The local search corresponds to the use of two operators that are directly applied to one or more routes from a solution that has been recreated in the precedent evaluation phase.

A solution for the HIRP-BS is composed of two parts, inventory and routing, as described below:

1. the inventory part that is defined by the heuristic procedure presented in Section 4.3.1 that allows the quantities placement at the very latest possible period (right shift) and can be optimal for the routing part since we consider the exact time period in which the inventory of customers is disrupted and also because these quantities can be anticipated through the Split **and**

2. the routing part that corresponds to the set of routes defined by the Split algorithm and their quality is totally dependent on the giant tour generation given by Section 4.3.1 from the quantities calculated, which means that an efficient local search should be applied as shown by Prins (2004) when applying a local search that alternates from the set of giant tours to the set of solutions in order to improve their quality.

Two types of neighborhood operators are used to enhance the quality of the solution: reinsertion and 2-OPT. These operators can be applied either within the same route (intra-route) or across different routes (inter-route). They specifically target improvements in the routing part of the problem. Moves involving customers assigned to different time periods are not considered, as they would require propagating delivery quantities across the entire time horizon, which adds significant complexity. Therefore, four neighborhood operators are utilized: reinsertion within routes and between routes, and 2-OPT within routes and between routes. An explanation, along with an illustration and the pseudo-algorithm, is provided below. For the illustrations, note that the customers placement represented by letters does not correspond to a geographic position and and are presented as follows to facilitate the comprehension of the moves considered.

For the sake of comprehension, a route R is identified by

$$R = \begin{cases} f(R) : \text{route cost (transportation + inventory)} \\ R.seq : \text{the sequence of customers to visit starting and ending by 0 (supplier)} \\ R.k : \text{vehicle type used} \\ f^{R.k} : \text{fixed cost of the vehicle type } R.k \text{ assigned} \\ v^{R.k} : \text{variable cost of the vehicle type } R.k \text{ assigned} \\ B^{R.k} : \text{capacity of vehicle type } R.k \text{ assigned} \\ R.load : \text{the amount of products that are carried by the vehicle} \\ R.lv_i : \text{quantity assigned to deliver customer } i \end{cases}$$

`2-OPT INTER`. Let R_1 and R_2 be two routes assigned to a period t from a feasible solution. Let also $a, b \in R_1$ connected by an arc e_{ab} of length $c_{a,b}$ and $c, d \in R_2$ two customers linked by an arc $e_{c,d}$ and distant of $c_{c,d}$ unities. This operator tests the removal of both arcs e_{ab} and $e_{c,d}$ and the creation of the arcs $e_{a,d}$ and $e_{d,b}$ in such a way that the beginning of R_1 until customer a is attached to the customers from d to the end of R_2 and that from the beginning of route R_2 until customer c is attached to customer b to the end of R_1. The corresponding procedure is presented in Algorithm 9.

Algorithm 9 has an input parameter the problem *data* given by Tables and 2.1 and 2.2 from the IRP and the HIRP-BS, respectively, a period t in which the search will be performed, two routes R_1 and R_2 and a maximum number of iterations $nLLS$. It starts by fixing route R_1 and exploring each position (lines 8 to 33) and varying all the positions in R_2 (lines 16 to 32). Since this operator consider testing the insertion of a part of a route into another, the cost variation is calculated by saving, for route R_1, the partial costs $\bar{c}_{R_1}, c_{R_1}$ in which $\bar{c}_{R_1}$ corresponds to the remaining cost once the arcs have been deleted and c_{R_1} to the new part that has been added from R_2 are calculated. The same occurs for $\bar{c}_{R_2}, c_{R_2}$.

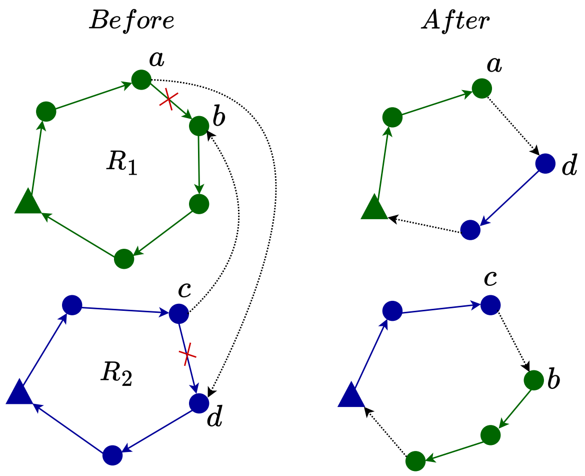


Figure 4.12: `2-OPT` inter routes

`2-OPT INTRA`. Let R be a route assigned to period t and $a, b, c, d \in R$ four customers such that a and b and connected by an arc e_{ab} and c and d by e_{cd}. This operator tests the removal of arcs e_{ab} and e_{cd} and adds the arcs e_{ac} and e_{bd}. Since it is an intra route operator, the set of customers belonging to R remains the same when an improvement

Algorithm 9: `2-OPT` inter routes

Data: *data*: problem data, t: a period, R_1, R_2: two routes, $nLLS$: maximum number of iterations

Result: R'_1, R'_2: two modified routes

Begin
$iterator \leftarrow 0$
$c_{R_1}, c_{R_2} \leftarrow 0$
$\bar{c}_{R_1} \leftarrow f(R_1)$
$\bar{c}_{R_2} \leftarrow f(R_2)$
$l_{R_1} \leftarrow 0$
$\bar{l}_{R_1} \leftarrow R_1.load$
for each position i in $R_1.seq$ **do**
  $n_1 \leftarrow R_1.seq_i$
  $n_2 \leftarrow R_1.seq_{i+1}$
  $c_{R_1} \leftarrow c_{R_1} + (c_{n_1,n_2} \cdot v^{R_2.k})$
  $\bar{c}_{R_1} \leftarrow c_{R_1} - (c_{n_1,n_2} \cdot v^{R_1.k})$
  $l_{R_1} \leftarrow l_{R_1} + R_1.lv_{n_1}$
  $\bar{l}_{R_1} \leftarrow l_{R_1} - R_1.lv_{n_1}$ $l_{R_2} \leftarrow 0$
  $\bar{l}_{R_2} \leftarrow R_2.load$
  for each position j in $R_2.seq$ **do**
    $n_3 \leftarrow R_2.seq_j$
    $n_4 \leftarrow R_2.seq_{j+1}$
    $c_{R_2} \leftarrow c_{R_2} + (c_{n_3,n_4} \cdot v^{R_1.k})$
    $\bar{c}_{R_1} \leftarrow c_{R_1} - (c_{n_3,n_4} \cdot v^{R_2.k})$
    $l_{R_2} \leftarrow R.lv_{n_3}$
    $\bar{l}_{R_2} \leftarrow \bar{l}_{R_2} - R.lvn_3$
    $\delta^{temp} \leftarrow (c_{n_1,n_4} - c_{n_3,n_4}) \cdot v^{R_2.k} + (c_{n_3,n_2} - c_{n_1,n_2}) \cdot v^{R_1.k}$
    if $\delta^{temp} < 0$ and $\delta^{temp} < \delta^{best}$ and $l_{R_1} + \bar{l}_{R_2} \leq B^{R_1.k}$ and $\bar{l}_{R_1} + l_{R_2} \leq B^{R_2.k}$ **then**
      `// improvement, save move` $\delta^{best} \leftarrow \delta^{temp}$
    end
    $iterator \leftarrow iterator + 1$
    if $iterator > nLLS$ **then**
      $R'_1 \leftarrow bestMove(R_1)$
      $R'_2 \leftarrow bestMove(R_2)$
    end
  end
end
`// perform best move in` R_1 `and` R_2
return R'_1, R'_2
End

is observed. Note that all the arcs from b to c need to be reversed only and only if the distance matrix is not symmetric, which is not the case of the HIRP-BS since for any two customers $i, j \in \mathcal{N}'$, $c_{ij} = c_{ji}$.

Algorithm 10: `2-OPT` intra routes

Data: *data*: problem data, t: a period, R: one route, $nMaxIt$: maximum number of iterations

Result: R': the modified route

Begin
$iterator \leftarrow 0$
$\delta^{best} \leftarrow \infty$
for each position i in $R.seq$ **do**
    $n_1 \leftarrow R.seq_i$
    $n_2 \leftarrow R.seq_{i+1}$
    $\delta^1 = -c_{n_1,n_2}$
    for $j = i + 2$ **do**
        $n_3 \leftarrow R.seq_j$
        $n_4 \leftarrow R.seq_{j+1}$
        $\delta^2 = c_{n_1,n_3} + c_{n_2,n_4} - c_{n_3,n_4}$
        $\delta = (\delta^1 + \delta^2) \cdot v^{R.k}$
        if $\delta < 0$ and $\delta < \delta^{best}$ **then**
            `// improvement, save move`
            $\delta^{best} \leftarrow \delta$
        end
        $iterator \leftarrow iterator + 1$
        if $iterator > nMaxIt$ **then**
            $R' \leftarrow bestMove(R)$
        end
    end
end
`// perform best move in` R
return R'
End

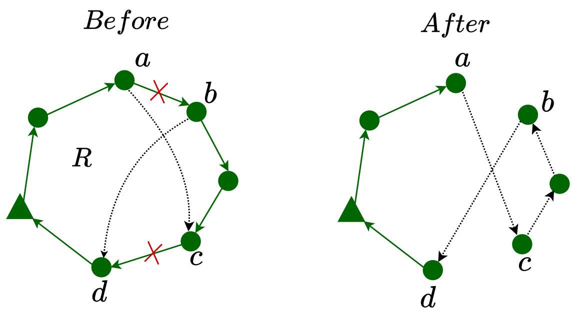


Figure 4.13: `2-OPT` intra routes

`INSERTION INTER`. Let R_1 and R_2 be two routes assigned to a period t. In R_1, three customers a, b and c connected by arcs e_{ab}, e_{bc} and for R_2, two customers d, e represented by an arc e_{de} are chosen. This operator tries to remove one customer from a route and try its insertion into another route. In Figure 4.14, the insertion of $b \in R_1$ is tested in R_2

by removing arcs e_{ab}, e_{bc} and e_{de} and adding arcs e_{db} and e_{be} in order to place customer b between d and e. Note that this operator may reduce the length of the route in which the customers are removed and it may also delete completely a route if all the customers could be better placed in the other one.

Algorithm 11: Insertion inter routes

Data: $data$: problem data, t: a period, R_1, R_2: two routes, $nMaxIt$: maximum number of iterations

Result: R_1', R_2': two modified routes

Begin
$iterator \leftarrow 0$
$\delta^{best} \leftarrow \infty$
for each position i in $R_1.seq$ **do**
 $n_1 \leftarrow R_1.seq_{i-1}$
 $n_2 \leftarrow R_1.seq_i$
 $n_3 \leftarrow R_1.seq_{i+1}$
 $\delta^1 = (c_{n_1,n_3} - c_{n_1,n_2} - c_{n_2,n_3}) \cdot v^{R_1.k}$
 if $R_1.lv_{n_2} + R_2.load \leq B^{R_2.k}$ **then**
  for each position j in $R_2.seq$ **do**
   $n_4 \leftarrow R_2.seq_j$
   $n_5 \leftarrow R_2.seq_{j+1}$
   $\delta^2 = (c_{n_4,n_2} + c_{n_2,n_5} - c_{n_4,n_5}) \cdot v^{R_2.k}$
   if $\delta^1 + \delta^2 < 0$ and $\delta^1 + \delta^2 < \delta^{best}$ **then**
    `// improvement, save move`
    $\delta^{best} \leftarrow \delta^1 + \delta^2$
   end
   $iterator \leftarrow iterator + 1$
   if $iterator > nMaxIt$ **then**
    $R_1' \leftarrow bestMove(R_1)$
    $R_2' \leftarrow bestMove(R_2)$
   end
  end
 end
end
`// perform best move in` R_1 `and` R_2
return R_1', R_2'
End

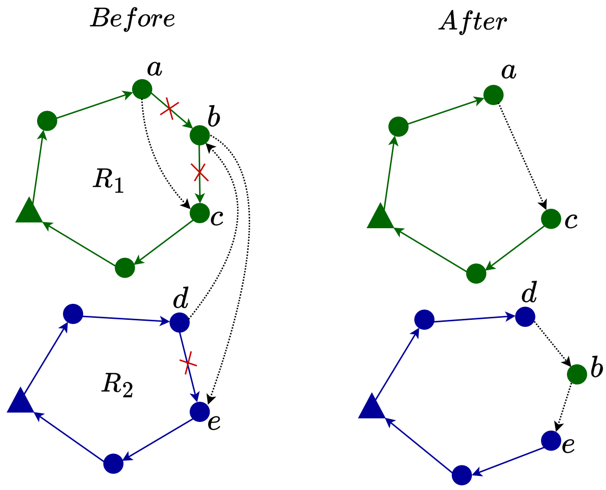


Figure 4.14: Insertion inter routes

INSERTION INTRA. Let R be a route in period t. The objective is to choose one customer and try to insert into the same route but between other customers. In Figure 4.15, five customers a, b, c, d and e are considered to try the insertion of b in another position of the route. The arcs e_{ab} that relies customers a and b, e_{bc} connecting b and c and e_{de} for d and e are removed and customer b is placed between d and e. For that, an arc e_{ac} is added from a to c and the two arcs e_{db} and e_{be} are added to insert b between d and e.

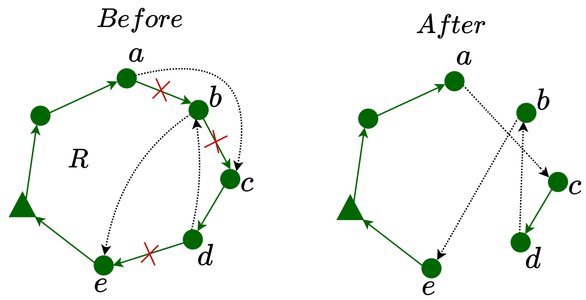


Figure 4.15: Insertion intra routes

Note that the 2-OPT operators involve reversing multiple arcs, which can increase the complexity of these moves based on the size of the routes being considered. In the method, distances are assumed to be symmetric between customers and the supplier, meaning that for any pair of (i, j), the costs $c_{i,j}$ and $c_{j,i}$ are identical.

The four operators described are integrated into the global adaptive framework given by Algorithm 13, where those operators that most significantly enhance the solution are given priority. Initially, each operator has an equal chance of being selected, i.e., $P_1 = P_2 = P_3 = P_4 = 25\%$. As the search progresses, if a move improves the current solution, its probability of being chosen in subsequent iterations is increased by a factor of δ. Consequently, the probabilities for the remaining three operators are decreased by $\frac{\delta}{3}$ units each. To prevent any single operator from dominating, lower (P^-) and upper (P^+) bounds for the probabilities P_o are enforced at each iteration, ensuring $P^- \leq P_o \leq P^+$ for

Algorithm 12: Insertion intra routes

Data: *data*: problem data, t: a period, R: one route, $nMaxIt$: maximum number of iterations

Result: R': the modified route

Begin
$iterator \leftarrow 0$
$\delta^{best} \leftarrow \infty$
for each position i in $R.seq$ **do**
    $n_1 \leftarrow R.seq_{i-1}$
    $n_2 \leftarrow R.seq_i$
    $n_3 \leftarrow R.seq_{i+1}$
    $\delta^1 = c_{n_1,n_3} - c_{n_1,n_2} - c_{n_2,n_3}$
    for $j = i + 1$ **do**
        $n_4 \leftarrow R.seq_j$
        $n_5 \leftarrow R.seq_{j+1}$
        $\delta^2 = c_{n_4,n_2} + c_{n_2,n_5} - c_{n_4,n_5}$
        $\delta = (\delta^1 + \delta^2) \cdot v^{R.k}$
        if $\delta < 0$ and $\delta < \delta^{best}$ **then**
            // `improvement, save move`
            $\delta^{best} \leftarrow \delta$
        end
        $iterator \leftarrow iterator + 1$
        if $iterator > nMaxIt$ **then**
            $R' \leftarrow bestMove(R)$
        end
    end
end
// `perform best move in` R
return R'
End

each operator $o \in \{1, \ldots, 4\}$. This dynamic probability adjustment system is inspired by the work of Chassaing et al. (2016). Finally, the adaptive global local search procedure is performed $nGLS$ times, with each local search operator allowed a maximum of $nLLS$ moves.

Algorithm 13: Global local search

Data: *data*: problem data, *nGLS*: global maximum number of iterations, *nLLS*: local maximum number of iterations *s*: current solution

Result: s': updated solution

Begin
$iterator \leftarrow 0$
while $iterator < nGLS$ **do**
    with a probability P_1 **do**
        Select one route R_1
        Apply `2-OPT_intra_route`$(data, t, R_1, nLLS)$
        if improvement **then**
            Save move on s'
            $P_1 = min(P_1 + \delta; P^+)$
            $P_{i,i\neq 1} = max(P_i - \frac{\delta}{3}; P^-)$
        else
            $P_1 = max(P_1 - \delta; P^-)$
            $P_{i,i\neq 1} = min(P_i + \delta; P^+)$
        end
    with a probability P_2 **do**
        Select two routes $R_1, R_2, R_1 \neq R_2$
        Apply `2-OPT_inter_routes`$(data, t, R_1, R_2, nLLS)$
        if improvement **then**
            Save move on s'
            $P_2 = min(P_2 + \delta; P^+)$
            $P_{i,i\neq 2} = max(P_i - \frac{\delta}{3}; P^-)$
        else
            $P_2 = max(P_2 - \delta; P^-)$
            $P_{i,i\neq 2} = min(P_i + \delta; P^+)$
        end
    with a probability P_3 **do**
        Select two routes $R_1, R_2, R_1 \neq R_2$
        Apply `Insertion_inter_routes`$(data, t, R_1, R_2, nLLS)$
        if improvement **then**
            Save move on s'
            $P_3 = min(P_3 + \delta; P^+)$
            $P_{i,i\neq 3} = max(P_i - \frac{\delta}{3}; P^-)$
        else
            $P_3 = max(P_3 - \delta; P^-)$
            $P_{i,i\neq 3} = min(P_i + \delta; P^+)$
        end
    with a probability P_4 **do**
        Select one route R_1
        Apply `Insertion_intra_routes`$(data, t, R_1, nLLS)$
        if improvement **then**
            Save move on s'
            $P_4 = min(P_4 + \delta; P^+)$
            $P_{i,i\neq 4} = max(P_i - \frac{\delta}{3}; P^-)$
        else
            $P_4 = max(P_4 - \delta; P^-)$
            $P_{i,i\neq 4} = min(P_i + \delta; P^+)$
        end
end
return s'
End

4.5 Post-optimization

The approach aims to improve a solution quality by exploring neighbors near the best solution identified by the ELS. To achieve this, it is necessary to define a "distance" metric based on the routes, which will then be used to recalculate the delivery quantities. This involves extending the linear models from Chapter 2 (Equations (2.1) through (2.18) and (2.19) through (2.40)), thereby initiating an iterative post-optimization process. The concept of distance allows for modifications in the routes, such as adding or removing arcs, and adjusting the previously calculated quantities for each customer and time period, all according to a specified degree of flexibility. The linearization process and the notion of distance is similar to what has been done in Chapter 3, Section 3.2.

4.5.1 Linearization constraints

Let $X = \bar{x}i, j^{k,t}$ denote the set of binary parameters representing the routes and their assignments to vehicles and time periods, while $Q = q_i^{k,t}$ represents the delivery quantities for each customer across different time periods. The values of $\bar{X}$ are derived from the binary variables $x_{i,j}^{k,t}$ obtained from the metaheuristic phase of SEMPO; specifically, if $x_{i,j}^{k,t} = 1$, then $\bar{x}_{i,j}^{k,t} = 1$, otherwise $\bar{x}_{i,j}^{k,t} = 0$. Both $\bar{X}$ and Q come from the optimal solution identified after the completion of the ELS iterations. Additionally, let $X = x_{i,j}^{k,t}$ represent the decision variables used in the linear models outlined in Chapter 2, and $\delta = \delta_{i,j}^{k,t}$ be a new set of binary variables that capture the absolute difference between each pair $x_{i,j}^{k,t}, \bar{x}_{i,j}^{k,t}$, as defined by Constraints (4.38). Constraints (4.39) are used to replace (4.38) because the latter is non-linear.

$$\delta_{ij}^{k,t} = |\bar{x}_{ij}^{k,t} - x_{ij}^{k,t}| \qquad \forall i, j \in \mathcal{N}', t \in \mathcal{T}, k \in \mathcal{K}^t \tag{4.38}$$

$$\eta = \begin{cases} \delta_{ij}^{k,t} \geq \bar{x}_{ij}^{k,t} - x_{ij}^{k,t} \\ \delta_{ij}^{k,t} \geq x_{ij}^{k,t} - \bar{x}_{ij}^{k,t} \\ \delta_{ij}^{k,t} \leq \bar{x}_{ij}^{k,t} + x_{ij}^{k,t} \\ \delta_{ij}^{k,t} \leq 2 - \bar{x}_{ij}^{k,t} - x_{ij}^{k,t} \end{cases} \qquad \forall i, j \in \mathcal{N}', t \in \mathcal{T}, k \in \mathcal{K}^t \tag{4.39}$$

4.5.2 Distance definition

To establish the degree of freedom, a single integer variable ϕ in Constraint (4.40) is introduced and denotes the allowable degree of freedom for modifying the current solution. Let ξ_i represent the maximum number of arc changes from the previous iteration, and Δ be the increment for the current iteration. Therefore, Constraint (4.41) is incorporated to restrict the number of arc changes permitted at each iteration.

$$\phi = \sum_{i \in \mathcal{N}'} \sum_{j \in \mathcal{N}'} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}^t} \delta_{i,j}^{t,k} \tag{4.40}$$

$$\xi_i \leq \phi < \xi_i + \Delta \tag{4.41}$$

□

The extended model incorporates the objective function and constraints detailed in (2.1)-(2.18) and (2.19)-(2.40), as well as (4.39) and (4.40), with iterative adjustments based on the progress of the search iterations as given by constraints (4.41).

Figure 4.16 depicts the post-optimization process. Search spaces are denoted by $\mathbb{Y}^i$, where i indicates each iteration of the procedure. The initial solution (the best ELS solution) is represented as s', and the best-improved solution at $\mathbb{Y}^i$ is denoted as s^i. In the first iteration, the delivery quantities Q are used as the starting point for the extended model, and ξ_1 is initialized to zero. Consequently, in the first iteration, Constraint (4.41) simplifies to $0 \leq \phi \leq \Delta$. After finding a new solution s^1, the ξ_i variables are updated to the value of ϕ from the previous iteration, adjusting Constraint (4.41) accordingly. The model is then resolved to find a potentially better solution. Thus, in the subsequent iteration, solutions are constrained by s^1. In the second iteration, Constraint (4.41) becomes $\xi_2 \leq \phi \leq \xi_2 + \Delta$, and after finding s^2, ξ_2 is updated to the new ϕ value. If no feasible solutions are found in the third iteration within $\mathbb{Y}^3$, the allowed deviation is doubled, adjusting Constraint (4.41) to $\xi_3 \leq \phi \leq \xi_3 + 2\Delta$. A new solution s^4 is then found in the fourth iteration.

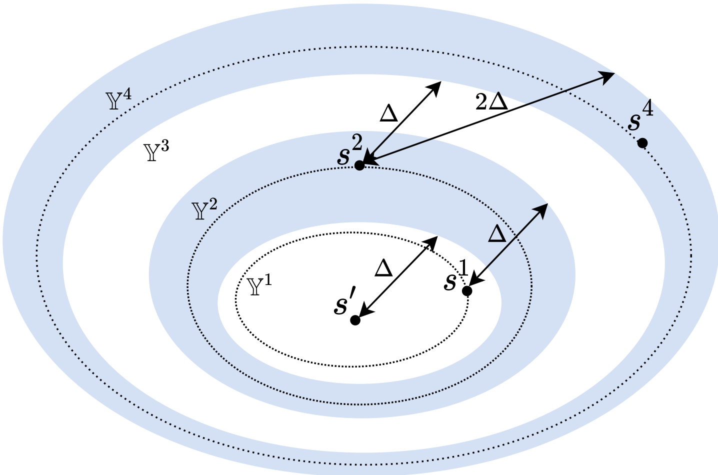


Figure 4.16: Post-optimization search schema

The algorithm iterates continuously until either a maximum allowable distance Δ^{max} from the solution s^G is reached or a maximum number of iterations, $pOPTMax$, is completed. Each iteration is constrained by a predefined time limit, $timepOPT$, set for the solver. It is important to note that the $\bar{X}$ values remain constant, ensuring that the search space remains centered around the initial solution s^G, with the space being constrained by s^i at each iteration.

Additionally, if at iteration i the zone defined by $\phi \leq \xi_{i-1} + \Delta$ does not yield any feasible solutions, exploring this region again is theoretically redundant. However, in practice, it provides a useful starting point based on the best solution found in the previous iteration. This approach allows for consideration of alternative values for the search space limits, thereby guiding the convergence of the algorithm in a different manner.

4.6 Experiments and results

This section presents the computational experiments to evaluate the SEMPO algorithm and to compare its performance with both the mathematical formulation detailed in Section 2.3 of Chapter 2 and the most recent algorithm introduced by Skålnes et al. (2023). The experiments were conducted on a system powered by an Intel Xeon Gold 6240R 2.40GHz processor with 256GB of RAM. Both the mathematical formulation and the

SEMPO algorithm were implemented in C++. The MILP formulation, along with the Post-optimization approach, was solved using the CPLEX MILP solver version 20.1.0.

A time limit of one hour was imposed for solving the proposed MILP formulation, including its linear relaxation, while the Post-optimization approach was given a time limit of 60 seconds. For the SEMPO algorithm, 10 runs were executed for each instance, with a computational time limit of 30 minutes per run. The parameters utilized for the SEMPO algorithm are provided in Table 4.1 and were chosen empirically based on multiple experiments.

Table 4.1: Metaheuristic parameters

Parameter	Value	Parameter	Value	Parameter	Value
λ	80%	$nGLS$	50	$[P-, P^+]$	$[10, 90]$
$timeSEMPO$	1800s	$nLLS$	200	δ	3
$iterELS$	5	$timeOPT$	60s	$\mathcal{L}^{max}$	10
$nNeighs$	10	Δ	3		
$nMut$	15	Δ^{max}	10		

The proposed new set of instances for the HIRP-BS was introduced in Section 2.4 of Chapter 2. Sections 4.6.1 and 4.6.2 analyze the computational results obtained from the mathematical formulation and the SEMPO method for the instances available in the literature and the newly generated benchmark instances presented in Subsection 2.4 of Chapter 2, respectively.

4.6.1 Results for the literature instances

The proposed SEMPO algorithm was designed to solve the HIRP-BS. This section presents the results obtained by the SEMPO approach when applied to the classic IRP, which represents a simpler variant of the HIRP-BS. While the proposed method yields satisfactory results for IRP instances, it is expected that specialized IRP methods available in the literature perform better.

Table 4.2 shows the results for a selection of instances proposed by Archetti et al. (2007) (as discussed in Subsection 2.4.1 of Chapter 2). These instances were randomly selected to demonstrate the metaheuristic performance in solving the IRP. The first three columns describe the instance configurations: the group of "abs" instances, the number of customers $|\mathcal{N}|$, the number of periods in the time horizon $|\mathcal{T}|$, and the number of vehicles (#Vehicles). The following two columns present the solution cost (z) and the running time in seconds (t) reported by Skålnes et al. (2023), which details the best-known solutions in the literature. In this column, values in bold represent the optimal costs reported by Archetti et al. (2012) and Manousakis et al. (2021) for the single and multiple-vehicle cases, respectively.

Subsequently, the table shows the cost of the best solution provided by the SEMPO algorithm across all runs (z^*) and the corresponding running time in seconds (t^*) required to obtain it. The final column provides the percentage gap between z^* and z.

According to Table 4.2, the SEMPO algorithm produces high-quality solutions for IRP instances, with an average gap of 2% from the best-known solutions in the literature, while requiring significantly less computational time (an average of 11 seconds compared to 33 minutes). Additionally, for the first instance, the optimal value is obtained (highlighted in

Table 4.2: Results on the high cost instances from Archetti et al. (2007)

Instance				Skålnes et al. (2023)		SEMPO algorithm		*Gap*
abs	$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	#Vehicles	z	t (s)	z^*	t^* (s)	
5	30	3	1	**9773.90**	1	**9773.90**	2	0.00
5	30	3	2	**10063.46**	110	10079.30	15	0.16
5	30	3	3	**10450.30**	72	10508.50	12	0.55
5	50	3	1	**15678.67**	84	16120.50	11	2.74
5	50	3	2	**16027.70**	7200	16361.90	18	2.04
5	50	3	3	16537.10	7198	17157.40	9	3.62
2	40	3	1	**11317.85**	612	11681.30	14	3.11
2	40	3	2	**11665.70**	1585	12078.70	15	3.42
2	40	3	3	12015.60	1223	12339.70	5	2.63
Avg.					2009		11	2.09

bold).

4.6.2 Results for the new benchmark instances

This subsection presents the results obtained by the MILP formulation and the SEMPO algorithm for the newly proposed set of instances introduced in Chapter 2, Subsection 2.4.2. The results are organized into three tables, each corresponding to a different subset of instances.

First, Table 4.3 shows the results for the small-scale instances. For each of the 13 instances, the initial columns show the results obtained by solving the MILP formulation, including: the cost of the optimal solution from the Linear Relaxation (LR), the Upper Bound (UB) at the conclusion of the MILP solver execution, the percentage gap (Gap_{UL}) between the UB and the Lower Bound (LB), and the running time in seconds to reach the UB (t^*).

The next five columns present the SEMPO algorithm results across all 10 runs: the cost of the best solution (z^*), the cost of the worst solution (z^w), the average cost ($\bar{z}$), the standard deviation (z^{sdv}) of the solutions obtained, and the average time across all runs to find the best solution ($\bar{t^*}$) in seconds. The final two columns provide the percentage gaps between z^* and the cost of the LR solution (Gap_{LR}), and between z^* and the UB obtained by the MILP formulation (Gap_{UB}). The last row of the table summarizes the average values.

No solution found by the proposed formulation, as solved by CPLEX, was proven to be optimal within the one-hour running time limit. Additionally, for the first two small-scale instances (instances 1 and 2), the MILP model failed to provide a feasible solution, whereas the SEMPO algorithm successfully generated solutions for all small-scale instances. The SEMPO algorithm outperformed the MILP formulation, providing better solutions for four instances (instances 4, 5, 10, and 11, highlighted in bold) in approximately half the execution time. For the remaining instances, the solution costs were, on average, very close to the best solutions obtained by both methods. The SEMPO algorithm achieved an average Gap_{UB} of 0.31%, with an average running time of 850 seconds to find the best solution—four times faster than the average running time of 3519 seconds for the proposed

Table 4.3: Results for the small-scale instances

Id.	MILP model				SEMPO algorithm					Gap_{LR}	Gap_{UB}
	LR	UB	Gap_{UL}	t^* (s)	z^*	z^w	$\bar{z}$	z^{sdv}	$\bar{t^*}$ (s)		
1	14362.21	-	-	3600	**15070.70**	15186.60	15134.38	39.25	789	4.70	-
2	14349.83	-	-	3600	**15430.10**	15554.10	15477.01	38.34	626	7.00	-
3	13192.78	**13787.07**	2.31	2755	14000.50	14188.70	14086.11	57.48	947	5.77	1.52
4	15117.77	16070.72	4.11	3600	**15860.80**	15945.00	15899.31	32.12	841	4.68	-1.32
5	12649.02	14871.81	12.26	3600	**14470.00**	14746.90	14575.43	93.03	722	12.58	-2.78
6	12013.27	**12709.82**	2.71	3600	13077.60	13188.20	13140.11	32.91	951	8.14	2.81
7	11091.08	**11667.72**	2.33	3600	11855.70	11948.10	11910.31	24.30	1063	6.45	1.59
8	13417.15	**13948.06**	1.96	3600	14497.40	14663.30	14583.48	63.23	546	7.45	3.79
9	11195.60	**11722.88**	1.98	3560	11953.50	11997.70	11979.08	12.65	994	6.34	1.93
10	12406.95	13327.03	4.31	3600	**13324.60**	13420.20	13374.03	34.05	970	6.89	-0.02
11	15287.85	17630.63	11.70	3600	**16263.50**	16363.60	16317.16	33.57	1245	6.00	-8.41
12	15118.76	**15689.13**	1.92	3562	16041.20	16142.80	16105.04	30.15	716	5.75	2.19
13	16058.64	**16535.69**	1.51	3600	16889.80	17046.40	16984.97	50.20	636	4.92	2.10
Avg			4.28	3519				41.64	850	6.67	0.31

formulation. Furthermore, the standard deviation of the costs among the 10 best solutions found by the SEMPO algorithm was sufficiently low to confirm the metaheuristic stability across different runs. The average cost of the best solutions provided by the SEMPO algorithm was 6.67% higher than the cost of the linear relaxation, as indicated by the Gap_{LR}.

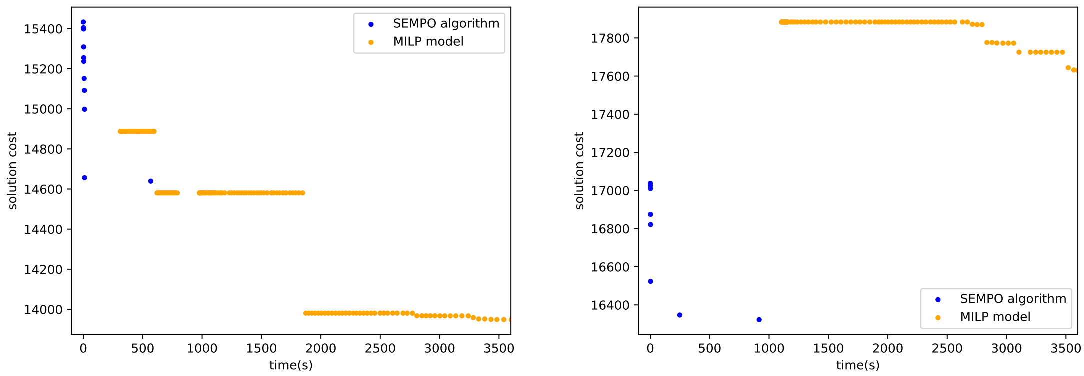
Additionally, Table 4.4 summarizes the results for the same small-scale instances, considering the ten runs performed by SEMPO. For each of the 13 instances, the table presents the gap (GapF) between the best Upper Bound (UBF) and Lower Bound (LBF) provided by CPLEX within the time limit for solving the proposed formulation. It also shows the best and worst runs, including the minimum, maximum, average, and standard deviation values of the gap (GapUB) between the cost of the best solution obtained by SEMPO (UBGE) and UBF. Similarly, the table presents the gap (GapLR) between UBGE and the linear relaxation of the formulation LR_F. The last row of the table provides the average gaps for all small-scale instances.

In Table 4.4, it is observed that the average gap between the MILP model and SEMPO is approximately 1%, with a very low standard deviation. Notably, for three instances (instances 4, 5, and 11), this gap is negative. When considering the linear relaxation, the gap is around 7%, also accompanied by a very low standard deviation.

Figure 4.17 illustrates the convergence of the SEMPO algorithm (for the best run) and the MILP model for two small-scale instances: one where the MILP model demonstrates the best convergence and one where it shows the worst convergence compared to the SEMPO algorithm.

Table 4.4: Gaps for the small-scale instances

ID	MILP model			SEMPO - Gap_{UB} (%)						SEMPO - Gap_{LR} (%)						$\bar{t}(s)_{SEMPO}$
	Gap_F(%)	UB	t(s)	best	worst	min	max	avg	sdv	best	worst	min	max	avg	sdv	
1	-	-		-	-	-	-	-	-	5.38	4.99	4.70	5.43	5.10	0.25	789
2	-	-		-	-	-	-	-	-	7.22	7.33	7.00	7.74	7.28	0.23	626
3	2.31	13787.07	2755	1.52	2.53	1.52	2.83	2.12	0.40	6.18	6.73	5.77	7.02	6.34	0.38	947
4	4.11	16070.71	3602	-0.92	-0.79	-1.32	-0.79	-1.08	0.20	4.71	5.19	4.68	5.19	4.92	0.19	841
5	12.26	14871.80	3606	-2.70	-0.85	-2.78	-0.85	-2.04	0.65	12.58	14.23	12.58	14.23	13.21	0.55	722
6	2.71	12709.81	3605	3.15	3.52	2.81	3.63	3.27	0.24	8.46	8.81	8.14	8.91	8.58	0.23	951
7	2.33	11667.71	3605	1.59	2.07	1.59	2.35	2.04	0.20	6.77	6.91	6.45	7.17	6.88	0.19	1063
8	1.96	13948.05	3603	4.72	4.40	3.79	4.88	4.36	0.42	8.37	8.04	7.45	8.50	8.00	0.40	546
9	1.98	11722.88	3560	1.93	2.29	1.93	2.29	2.14	0.10	6.49	6.69	6.34	6.69	6.54	0.10	994
10	4.31	13327.03	3603	0.49	0.47	-0.02	0.69	0.35	0.25	7.20	7.34	6.89	7.55	7.23	0.24	970
11	11.70	17630.62	3600	-8.01	-7.91	-8.41	-7.74	-8.05	0.22	6.20	6.43	6.00	6.57	6.31	0.19	1245
12	1.92	15689.12	3562	2.60	2.77	2.19	2.81	2.58	0.18	5.99	6.31	5.75	6.34	6.12	0.18	716
13	1.51	16535.68	3607	2.75	2.10	2.10	3.00	2.64	0.29	5.65	4.92	4.92	5.79	5.45	0.28	636
Avg	4.28			0.65	0.96	0.31	1.19	0.76	0.29	7.02	7.22	6.67	7.47	7.07	0.26	



(a) ID 8 - best convergence (b) ID 11 - worst convergence

Figure 4.17: Small-scale instances SEMPO convergence

Figure 4.17a illustrates the case where the MILP model shows the best convergence compared to the SEMPO algorithm. During the first 500 seconds, the SEMPO algorithm provides a better solution than the MILP model. However, in this specific instance, the metaheuristic gets trapped in a local minimum, and by the end of the one-hour computational time, CPLEX finds a better solution. In contrast, Figure 4.17b demonstrates a scenario where the SEMPO algorithm initially provides a solution of significantly better quality than the first solution found by CPLEX using the MILP model. The SEMPO algorithm then maintains a high convergence rate, ultimately yielding a solution with a significantly lower cost than that found by CPLEX by the end of the execution.

Table 4.5 presents the results for the medium-scale instances. For the MILP model, it shows the cost obtained from the linear relaxation. For the SEMPO algorithm, it lists the cost of the best solution (z^*), the cost of the worst solution (z^w), the average cost across all runs ($\bar{z}$), the standard deviation of the costs across all runs (z^{sdv}), and the average time across all runs to find the best solution ($\bar{t^*}$) in seconds. Unlike Table 4.3, only the cost of the linear relaxation (column LR) is provided here, as the MILP formulation failed to find a feasible solution for all medium-scale instances. Consequently, only the gap between the best solution provided by the SEMPO algorithm and the linear relaxation cost (Gap_{LR}) can be calculated. For instances 1 and 3, which have 83 customers and a time horizon of 28 periods, the Gap_{LR} could not be calculated since the CPLEX solver was unable to optimally solve the linear relaxation within the one-hour time limit. The last row of the table provides the average values.

In Table 4.5, for the medium-scale instances, the average Gap_{LR} is 8.53%. On average, the SEMPO algorithm finds the best solution in less than 20 minutes (921 seconds). These results are considered to be of high quality given the complexity of the HIRP-BS compared to the classical IRP. The difficulty of solving the HIRP-BS stems from its characteristics, such as a heterogeneous fleet of vehicles, variable costs and demands, batch sizes per customer, etc., as well as the size of the time horizon and number of customers. This is evident from the fact that even for instances with 19 customers and a time horizon of 14 periods, the CPLEX solver fails to find feasible solutions for the proposed model within the one-hour computation time.

Table 4.5: Results for the medium-scale instances

Instances			MILP model	SEMPO algorithm					Gap_{LR}
$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{T}\rvert$	Id.	LR	z^*	z^w	$\bar{z}$	z^{sdv}	$\bar{t^*}$ (s)	
20	14	1	31765.05	33861.00	34310.40	34135.72	152.40	568	6.19
		2	35995.55	38037.40	38184.90	38098.06	50.75	934	5.37
		3	36849.88	39391.00	39677.50	39533.20	87.06	845	6.45
	21	1	70705.51	74315.90	74685.10	74477.94	125.89	634	4.86
		2	67279.77	71637.70	71959.80	71773.83	86.61	697	6.08
		3	81136.08	85644.50	86204.60	85957.79	213.05	1069	5.26
	28	1	105572.10	112247.97	114155.00	113299.10	813.64	1213	5.95
		2	108853.94	115040.73	116761.00	115569.96	486.08	723	5.38
		3	96870.66	102777.00	104181.00	103317.85	482.34	748	5.75
34	7	1	21504.21	23272.60	23462.20	23372.28	64.95	928	7.60
		2	25220.85	26918.00	27388.40	27168.72	138.79	843	6.30
		3	19551.37	21610.70	21773.50	21699.36	66.91	634	9.53
	14	1	64725.84	68249.90	68664.50	68480.09	148.40	1175	5.16
		2	58026.87	63853.30	64305.60	64156.87	152.21	471	9.12
		3	60027.36	66903.30	69413.60	67992.80	749.34	945	10.28
	21	1	112811.34	120413.00	121767.00	121000.50	449.45	788	6.31
		2	142540.14	152232.00	154337.00	153684.00	812.55	1019	6.37
		3	125961.65	135518.00	137021.00	136466.10	484.19	941	7.05
	28	1	199983.60	210551.00	211847.00	211313.40	379.20	948	5.02
		2	194899.22	211936.00	213026.00	212255.30	303.29	914	8.04
		3	198494.93	212032.48	214014.00	213063.18	730.89	1227	6.38
	7	1	28952.92	31531.70	32216.50	31836.35	186.75	1070	8.18

		2	31393.57	34461.50	35002.90	34739.92	179.66	772	8.90
		3	25408.63	27565.80	27845.20	27684.01	79.09	833	7.83
		1	70858.72	74801.30	75545.10	75232.20	203.08	1058	5.27
	14	2	85473.88	91480.70	91840.00	91686.45	103.67	802	6.57
		3	78769.42	85545.30	86500.70	86094.61	267.94	850	7.92
		1	163482.68	174044.00	174392.00	174175.90	99.85	945	6.07
	21	2	165078.24	173148.00	174971.00	174330.50	679.94	1088	4.66
		3	129226.17	138930.00	139341.00	139166.00	108.66	888	6.98
		1	232277.48	245113.00	245555.00	245387.30	148.83	888	5.24
	28	2	263986.87	281252.00	282505.00	281903.30	396.04	936	6.14
		3	249555.84	265153.00	265668.00	265435.40	191.95	1053	5.88
		1	46315.26	52713.20	53387.20	53026.60	251.08	676	12.14
	7	2	27474.70	33768.10	34370.00	34107.19	194.95	1036	18.64
		3	37506.27	43614.60	44267.90	44028.97	242.42	784	14.01
		1	100262.00	112607.00	113317.00	112918.60	228.33	950	10.96
	14	2	102989.70	124007.00	124892.00	124339.50	266.90	728	16.95
58		3	118288.86	129879.00	132119.00	131319.70	847.64	1164	8.92
		1	199673.24	222510.00	223928.00	222944.20	479.55	1308	10.26
	21	2	198632.42	220827.00	222426.00	221688.20	527.22	910	10.05
		3	187080.90	216367.00	217453.00	216936.00	383.59	774	13.54
		1	312777.77	338614.00	341230.00	340238.10	831.13	993	7.63
	28	2	289651.41	324261.00	325817.00	324912.10	505.19	1091	10.67
		3	344415.47	380963.00	385579.00	382520.00	1399.66	822	9.59
		1	46176.09	53916.50	54359.30	54054.96	132.68	801	14.36
	7	2	45115.89	53376.90	53819.60	53657.34	124.79	679	15.48
		3	48785.68	55261.00	55892.40	55641.00	211.91	997	11.72
		1	150855.25	168151.00	168906.00	168624.30	251.30	874	10.29
	14	2	115554.71	129975.00	131003.00	130494.40	370.69	812	11.09
83		3	161514.81	179839.00	180555.00	180194.70	214.70	811	10.19
		1	220407.99	240328.00	242961.00	241556.80	713.51	1225	8.29
	21	2	272141.84	298430.00	301140.00	299549.70	826.73	1095	8.81
		3	254559.93	280787.00	282879.00	281799.80	659.61	1372	9.34
		1	–	485412.00	488434.00	487275.20	924.85	1044	–
	28	2	458910.43	499365.00	501946.00	500579.90	798.14	863	8.10
		3	–	442948.00	444221.00	443590.80	420.45	1250	–
Avg							375.97	921	8.53

Table 4.6, similar to the table for small-scale instances, provides the average gap across all ten runs of SEMPO for the medium-scale instances. In this case, there is no gap from an Upper Bound since no feasible integer solution was found by the MILP model. Therefore, only the gap from the linear relaxation is presented. Generally, these gaps are around 7%, similar to the small instances, indicating a certain convergence stability of the algorithm, a very low standard deviation.

Table 4.6: Gaps for the medium-scale instances

Instance		$\text{Gap}_{LR}(\%)$						Instance		$\text{Gap}_{LR}(\%)$					
$\mathcal{N}'$	$\mathcal{T}$	best	worst	min	max	avg	sdv	$\mathcal{N}'$	$\mathcal{T}$	best	worst	min	max	avg	sdv
19								34	7	8.24	8.73	7.82	8.73	8.35	0.34
	14	6.33	6.14	6.05	6.72	6.42	0.25		14	8.64	9.50	8.58	9.50	8.91	0.46
	21	5.66	5.82	5.43	5.82	5.65	0.17		21	7.46	6.81	6.81	7.65	7.24	0.39
	28	5.98	6.79	5.72	7.08	6.29	0.50		28	6.74	6.76	6.65	6.98	6.79	0.21
	Avg	5.99	6.25	5.97	6.40	6.12	0.31		Avg	7.77	7.95	7.62	7.96	7.82	0.35
46	7	8.91	9.11	8.59	9.48	8.97	0.42	58	7	15.63	15.38	15.29	15.84	15.64	0.45
	14	6.93	7.01	6.86	7.24	7.03	0.22		14	12.38	12.70	12.38	13.06	12.77	0.31
	21	6.26	6.17	5.98	6.31	6.20	0.17		21	11.45	11.55	11.45	11.71	11.53	0.19
	28	5.88	5.88	5.85	5.99	5.89	0.09		28	9.60	9.77	9.42	9.77	9.63	0.23
	Avg	7.00	7.04	6.94	7.13	7.02	0.22		Avg	12.27	12.35	12.27	12.50	12.39	0.29
83	7	14.35	14.37	14.01	14.51	14.27	0.25								
	14	10.74	10.79	10.65	10.92	10.78	0.16								
	21	9.21	9.32	9.04	9.34	9.19	0.24								
	28	7.97	7.91	7.78	7.98	7.88	0.14								
	Avg	10.57	10.60	10.45	10.60	10.53	0.20								
Global		8.86	8.97	8.86	8.97	8.92	0.27								

Lastly, Table 4.7 presents the results for the ten large-scale instances, similar to the information provided in the previous tables. For these large-scale instances, only the first two phases of the SEMPO algorithm were executed (as outlined in Figure 4.1), since including the third phase would significantly increase the computational time. For the last four instances in the table, the Gap_{LR} could not be calculated because the CPLEX solver was unable to optimally solve the linear relaxation of the model within the one-hour time limit. For the remaining instances, the average Gap_{LR} is 12.01%. Additionally, the total average execution time is less than 15 minutes.

It's also observed that, even though the post-optimization phase was not executed for the large-scale instances, the gap between the cost of the solutions provided by the SEMPO algorithm and the cost of the linear relaxation of the model is not significantly larger compared to the small and medium-scale instances where the full method was applied. This indicates that the SEMPO approach is efficient in terms of execution time and can still deliver high-quality solutions, even when the linear relaxation solved by CPLEX does not yield an optimal solution within a considerably longer running time.

Table 4.8 summarizes the gaps obtained for the large-scale instances across each run. For the last four instances, marked with an asterisk, the linear relaxation of the proposed model was not fully resolved within the one-hour time limit. This explains the occurrence of negative gaps for some of these instances.

For further analysis, the detailed table results are presented in Appendix C. Also, the convergence graphics for the small-scale instances comparing both the MILP model and the SEMPO algorithm performance is in Appendix D.

Table 4.7: Results for the large-scale instances

Instances			MILP model	SEMPO algorithm					Gap_{LR}
$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	Id.	LR	z^*	z^w	$\bar{z}$	z^{sdv}	$\bar{t^*}$ (s)	
114	7	1	63214.61	73697.50	74914.70	74120.87	357.73	1288.13	14.22
		2	62803.88	72296.70	73236.40	72902.97	293.43	1057.53	13.13
		3	62229.09	70788.60	71175.00	70952.09	129.24	836.96	12.09
	14	1	160846.19	183068.00	184650.00	183790.90	524.65	689.14	12.14
		2	240633.57	268914.00	271892.00	270369.50	965.68	809.31	10.52
	21	1	368429.18	409142.00	410697.00	409722.40	627.68	683.57	9.95
149	28	1	-	900932.00	905981.00	904061.50	2204.26	780.12	-
170	21	1	-	537235.00	539239.00	538574.90	687.44	1103.05	-
	28	1	-	852666.00	856599.00	854943.00	1628.57	888.50	-
183	7	1	-	98708.70	99711.30	99123.17	326.58	806.78	-
Avg							774.53	894.31	12.01

Table 4.8: Gaps for the large-scale instances

Instances			SEMPO		Gap_{LR}									
$\|\mathcal{N}'\|$	$\|\mathcal{T}'\|$	ID	UB	t(s)	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
114	7	1	63214.61	1288	14.66	14.22	14.66	14.60	14.91	14.69	15.07	15.62	14.33	14.36
114	7	2	62803.88	1058	13.83	14.13	14.03	13.13	13.56	13.51	13.97	14.00	14.24	14.11
114	7	3	62229.09	837	12.38	12.16	12.51	12.28	12.17	12.33	12.31	12.09	12.57	12.14
114	14	1	160846.19	689	12.64	12.14	12.39	12.39	12.64	12.14	12.64	12.29	12.66	12.89
114	14	2	240633.57	809	10.95	11.33	10.97	10.98	10.52	10.60	11.50	11.14	10.73	11.25
114	21	1	368429.18	684	9.95	10.07	10.29	9.97	10.16	9.95	9.97	10.29	10.16	9.97
149	28	1*	704377.65	780	22.17	22.25	22.25	22.25	22.17	22.17	21.82	21.82	22.17	21.82
170	21	1*	612220.46	1103	-13.55	-13.53	-13.68	-13.96	-13.85	-13.58	-13.62	-13.79	-13.61	-13.56
170	28	1*	688814.62	888	19.35	19.27	19.22	19.59	19.57	19.59	19.33	19.57	19.28	19.55
183	7	1*	326477.91	807	-227.42	-227.90	-230.07	-228.68	-230.46	-229.88	-229.63	-229.13	-229.77	-230.75
Global $\overline{Gap}$(%)				894	-12.51	-12.59	-12.74	-12.74	-12.86	-12.85	-12.66	-12.61	-12.72	-12.82

4.7 Chapter conclusion

This chapter addressed an algorithm to solve the new variant of the Inventory Routing Problem named Heterogeneous Inventory Routing Problem with Batch Size (HIRP-BS) which contains some characteristics that represent a more realistic scenario including a heterogeneous fleet of vehicles and a batch size per customer. In order to solve the problem, the mathematical formulation presented in Subsection 2.3 from Chapter 2 is used. The solution method introduced consists of a Split-based algorithm using a multi-period giant tour and a post-optimization search phase embedded in a metaheuristic, referred to as the SEMPO algorithm, to solve the problem.

To test the SEMPO, the new set of benchmarking instances introduced in Section 2.4 from Chapter 2 is used as well as the classical existing literature instances from Archetti et al. (2007) and Coelho et al. (2012a).

The results show that the proposed flow formulation solved by the CPLEX solver is not able to provide feasible solutions to the problem in a reasonable computational time for medium and large-scale instances. In addition, the model only provides feasible solutions to some of the small-scale instances. However, the SEMPO metaheuristic is able to find feasible solutions for all the proposed instances with better or competitive costs in a short running time.

Chapter remainder

- *A Split-based algorithm adapted to the HIRP-BS*
- *A metaheuristic schema which embeds the Split algorithm*
- *Computational experiments on the new benchmarking instances for the HIRP-BS*
- *Convergence analysis on the algorithm performance compared to the mathematical formulation*

Conclusion

This thesis addressed the integrated inventory management and routing problem named Inventory Routing Problem (IRP) as well as a new extension named Heterogeneous Inventory Routing Problem with Batch Size (HIRP-BS). The first stands for the classical version of the problem and has received a lot of attention of the OR community so far. The second was introduced in this thesis and considers an extension of the classical IRP in which some characteristics are added, such as a heterogeneous vehicle fleet, inventory holding costs and demands that are period-dependent as well as an inventory policy based on the delivery by batches rather than in single unities.

For both problems, the objective is to serve a set of customers that has demands over a finite horizon of time and consider a supplier from which a set of vehicles is scheduled at each period to perform the routes respecting a set of contraints. These include the vehicles capacity, the fact that a route starts and ends at the supplier and the inventory management, for example. The IRP and the HIRP-BS are challenging in terms of logistics operations in a supply chain and must be coordinated to avoid customers inventory disruption and to keep a satisfying level of service for both customers and supplier.

In order to do so, three approaches have been proposed: a mathematical formulation in Chapter 2, an iterative algorithm in Chapter 3 and a metaheuristic in Chapter 4. Below, a summary on these approaches and results is presented including the problem description and literature review presented in Chapter 1.

- In **Chapter 1**, the Inventory Routing Problem has been introduced and explained in terms of the inventory management problem and the routing problem associated. Multiple works from the literature have been highlighted to present what has already been proposed as extensions of the base classical problem and the methods and algorithms to solve them. In these literature papers, the instances provided are, in the majority of the cases, derived from the same classical set introduced in 2007, which does not incorporate enough features for the IRP variants. Also, from the bibliography review presented in this chapter, it is clear that more work can be done to get closer to a real scenario and the variation proposed (HIRP-BS) is of interest of the OR community since the features considered have never been proposed before.
- **Chapter 2** formalizes both IRP and HIRP-BS problems with the corresponding mathematical formulation and the set of instances proposed to solve each one. For the HIRP-BS, since it first appear in this theses, a new set of instances to match its characteristics is proposed. It is clear from the experiments presented that another strategies of resolution rather than the linear approaches are requested to obtain feasible upper bounds in a reasonable time in order to increase the convergence time since the number of variables for the HIRP-BS is very huge and the problem dimension increases fast considering the customers, periods and vehicles indexation.

Thus, it is impossible to provide a feasible integer solution for more than half of the new instances considered.

- In **Chapter 3**, another approach to solve the IRP was presented. It consists of an iterative algorithm that decomposes the initial problem into subproblems according to the number of periods considered. These subproblems are treated and solved sequentially according to the ascending period order and until all the periods have been explored. Each subproblem carries information from the previous ones to accelerate the algorithm convergence. They also consider a parameter responsible for defining the number of changes authorized so that the changes in the solutions are controlled according to this parameter to define a notion of neighborhood around the current solution. The results have shown promising solutions in terms of solution quality and convergence.

- **Chapter 4**, known to be the core chapter of this thesis, addresses a metaheuristic with a Split algorithm embedded to solve the HIRP-BS and has also been tested on the IRP. The metaheuristic can be seen as an extension of the Grasp×ELS that exists in the literature in which additional steps and features have been added to contemplate the problem treated. The algorithm relies on a three-step constructive heuristic that considers an evaluation phase (Split), a local search mechanism (ELS) that also incorporates the Split and a post-optimization. The first consists of defining a feasible solution for the problems through the evaluation of a sequence of customers named giant tour and splitting it into feasible routes. These are later improved by the ELS phase which also contains a mutation operator to provide diversification on the solutions found so far and to avoid local minima. Lastly, the post-optimization aims to use the mathematical formulation of the problems to improve the solution quality in terms of routing and inventory management. Several experiments are considered including the new set of instances and the results show a fast convergence of the method and promising solution quality compared to the linear formulation resolution. Different from the previous chapter, the metaheuristic is capable of finding one upper bound for all the new set of instances considered. Also, the metaheuristic present an insight on how its structure can be adapted to treat another types of problems and produce good quality solutions.

If the major contributions and the originality of this thesis were to be highlighted, the following aspects could be mentioned:

- Introduction of an IRP extension (the HIRP-BS) that considers a period-dependent heterogeneous vehicle fleet, demands and inventory costs and a batch size per customers;
- Introduction of a new set of instances to address the HIRP-BS features;
- A mathematical formulation of the HIRP-BS to handle the characteristics considered as an extension of the classical IRP;
- A Split algorithm capable of generating feasible solutions from a multi-period giant tour strategically defined to handle the customers demands over the periods and the constraints associated;

- An extension of the classical Grasp×ELS metaheuristic by incorporating the Split algorithm for the IRP and a post-optimization phase using the problem linear formulation and a commercial solver;

- An iterative decomposition algorithm for the IRP that divides the problem into subproblems that are solved sequentially and that reuses information from previous iterations to accelerate convergence.

As with all research problems, there is always place for further work to add more features and explore additional characteristics of the IRP, including new constraints and instance sets. Some perspectives are listed below.

First of all, the iterative approach presented for the classical IRP version can be extended for the HIRP-BS to analyse and compare its performance with the linear formulation proposed as well as the SEMPO metaheuristic. Also, the development of a Branch-and-Cut algorithm could be useful to help the HIRP-BS mathematical formulation in generating feasible solutions since it has been widely applied to the classical IRP version because the base formulations are also hard to solve. This B&C algorithm could also be incorporated into the iterative algorithm to increase the method performance.

As for new characteristics of the problem, we could consider the multi-product scenario, in which the linear formulations as well as the SEMPO metaheuristic complexity would be significantly increased. Alternatively, we could consider the fact that some products cannot be transported by some vehicles, which would help to reduce the number of possibilities for assignment and, consequently, the algorithm complexity.

Also, the use of artificial intelligence techniques could offer the methods presented in this thesis significant improvements in terms of finding promising solutions. The idea would be to analyse an extensive number of solutions for the problem and to identify which features contribute to designing good solutions and bad ones. With this information in hand, it would be insightful to incorporate strategies that can favor the prevalence of good solutions and avoid the bad ones.

Along with this thesis, the algorithms presented here can provide valuable insights for developing further extensions and adapting them to incorporate new features and address other OR problems. Not limited to IRPs, the iterative approach, the SEMPO metaheuristic, and the underlying ideas of these algorithms can also serve as inspiration for application to other OR problems.

Publications

This thesis has led to several publications, including national and international communications, a book chapter and a journal paper currently under review. Their references and main contributions are listed below.

- **International communications**

  1. **Martino, D.**, Lacomme, P., Farias, K., & Manuel, I. (2022, July). A Split-based Dynamic Programming approach for the Inventory Routing Problem. In EURO 2022-Espoo.

     This resume was orally presented in the International communication on the European Conference of Operational Research in Espoo, Finland, from july 3th to july 6th, 2022. It addresses a preliminary work on a Split algorithm developed to solve the IRP.

  2. **Martino, D. P.**, Lacomme, P., Farias, K., & Iori, M. (2023, April). A metaheuristic schema for the Inventory Routing Problem. In EU/ME meeting x Quantum School: Emerging optimization methods: from metaheuristics to quantum approaches.

     This paper is issued from the EU/ME international conference held in Troyes, France, from april 17th to april 21th, 2023. It addresses a preliminary work on a GRASPxELS metaheuristic to solve the IRP by using a Split-based algorithm adapted from the VRP literature.

- **National communications**

  1. Lucas, F., **Martino, D.**, Billot, R., & Lacomme, P. (2023, February). Inventory Routing Problem et Fouille de données: quel apport des règles de décision?. In ROADEF 2023: 24ème édition du congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision.

     This 2-pages resume in french was orally presented in the 24^{th} edition of the French conference on Operational Research in Rennes, France, from february 20th to february 23th, 2023. It is issued from a collaboration with Flavien Lucas and Romain Billot from the IMT Atlantique to apply data mining techniques to extract good and bad characteristics to classify the solutions. The study was carried on thousands of solutions of variate quality and allow the definition of decision rules based on these.

  2. **Perdigão, D.**, Lacomme, P., de Araújo, K. F., & Iori, M. (2023, February). Un algorithme basé sur la Programmation Dynamique pour l'Inventory Routing

Problem. In 24ème Congrès Annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision.

This 2-pages resume in french was orally presented in the 24^{th} edition of the French conference on Operational Research in Rennes, France, from february 20th to february 23th, 2023. It addresses the Split-based dynamic programming algorithm to solve the IRP as well as some local search operators acting on the routing part of the problem.

- **Book chapter**

    1. [1] Farias, K., Lacomme, P., & **Martino, D. P.** (2024, June). Iterative Heuristic over Periods for the Inventory Routing Problem. In Metaheuristics International Conference (pp. 123-135). Cham: Springer Nature Switzerland.

        This book chapter is published in Lecture Notes in Computer Science (LNCS) by Springer and is issued from the 15th Metaheuristics International Conference held in Lorient, France, from June 4th to June 7th, 2024. The paper addresses an iterative algorithm to solve the IRP which decomposes the mathematical formulation into subproblems according to the number of periods of time considered. Each i^{th} subproblem is treated as period-dependent problem and contains information about the previous ones according to a degree of freedom established. The algorithm is solved sequentially and aims to accelerate the convergence of the mathematical formulation that models the problem. Results confirm the fast convergence in terms of time, which reduces the decision-making time in some cases.

- **Journal**

    1. A Split-Embedded Metaheuristic for the Heterogenous Inventory Routing Problem with Batch Size. **Under review**.

        This paper is currently under review in a journal. It addresses a new variant of the classical Inventory Routing Problem by incorporating new characteristics and constraints such as a batch size per customer, period-dependent inventory holding costs, customers demands and an heterogeneous vehicle fleet as well as non-Euclidian distances among the customers and the supplier. This variant is then names the Heterogeneous Inventory Routing Problem with Batch Size (HIRP-BS). To handle this problem, a mathematical formulation which is an adaptation of an existing flow formulation is presented as well as a metaheuristic with a post-optimization phase. The metaheustics embeds a Split-based algorithm which consists in splitting a sequence of customers named giant tour into routes that are assigned to the set of available vehicles. To test the algorithm, the classical instances set from the literature is used and a new one is introduced to match the HIRP-BS characteristics. Several experiments are conducted and the convergence analysis is done in order to validate the metaheuristic approach and its capability of addressing this new problem variation. Results shown that the metaheuristic performs well and provides valid upper bounds on instances for which the mathematical formulation is not capable of doing so.

[1]Alphabetical family name authorship order

Résumé étendu

Introduction

La gestion des transports vise à choisir le mode le plus adapté parmi le rail, la mer, l'air et la route, en cherchant à minimiser les coûts tout en garantissant fiabilité et sécurité. Elle doit également faire face à plusieurs défis majeurs, tels que la congestion du trafic, les limitations des infrastructures, le coût des carburants, la pollution, la sécurité ou encore les aspects environnementaux. Pour surmonter ces difficultés, une planification stratégique et une collaboration entre les différents acteurs et secteurs concernés sont nécessaires.

D'autre part, la gestion des stocks se concentre sur le maintien de niveaux suffisants pour répondre aux demandes des clients sans rupture de stock, tout en assurant un service de qualité. Cette tâche est complexe, car elle doit relever plusieurs défis, notamment la prévision de la demande, la gestion des coûts, le cycle de vie des produits, la prévention des pertes et le maintien d'un équilibre précis des stocks pour éviter les ruptures ou les excès de stock.

Afin de gérer simultanément les problèmes de transport et de gestion des stocks, les techniques de Recherche Opérationnelle (RO) peuvent être appliquées. Le problème qui consiste à traiter à la fois la gestion des stocks et des transports est appelé Inventory Routing Problem (IRP) dans la littérature. Il a suscité l'intérêt de nombreux chercheurs au cours des dernières décennies en raison de sa complexité et de sa pertinence pratique. L'IRP a été introduit en 1983 par Bell et al., et comme pour la plupart des problèmes de RO, l'objectif est de trouver une solution optimale à coût minimal. Dans ce cas, il s'agit d'optimiser à la fois les coûts liés aux stocks et ceux liés au transport.

L'IRP considère un ensemble de clients et un fournisseur. Les clients ont des demandes déterministes, tandis que le fournisseur dispose d'une capacité de production sur un horizon temporel discret. Les opérations de réapprovisionnement doivent être planifiées de manière à éviter toute rupture de stock. Pour ce faire, un ensemble de véhicules à capacité limitée est disponible à chaque période, et les itinéraires desservant les clients doivent être définis.

En plus de traiter le problème classique de l'IRP, cette thèse propose une nouvelle extension de l'IRP en intégrant des caractéristiques intrinsèques aux systèmes réels, afin d'apporter des éléments de réponse aux trois questions précédentes. Cette extension est nommée *Heterogeneous Inventory Routing Problem with Batch Size* (HIRP-BS). Trois éléments sont ajoutés au modèle classique de l'IRP : une flotte de véhicules hétérogène et dépendante du temps, des livraisons par lots, ainsi que des demandes et des coûts de stockage non statiques selon les périodes. Ces trois aspects ont soit été peu explorés, soit n'ont pas reçu une attention suffisante dans la littérature jusqu'à présent.

La thèse est divisée en quatre chapitres, et les sections suivantes résument les techniques employées ainsi que les résultats obtenus dans chaque chapitre. La Section 1 présente l'IRP, en détaillant les opérations de gestion des stocks et de transport, ainsi qu'une revue de la littérature. La Section 2 formalise l'IRP et l'HIRP-BS en termes mathématiques, avec les variables, la fonction objective et les contraintes, et introduit deux ensembles d'instances : le classique et celui proposé dans cette thèse. Les résultats sont également discutés.

La Section 4, central à cette thèse, présente la métaheuristique basée sur l'algorithme de type Split pour résoudre l'IRP et l'HIRP-BS. Le processus de découpage en trois étapes est décrit, suivi des mécanismes de mutation et de recherche locale pour optimiser le transport, ainsi qu'une phase de post-optimisation pour le transport et les stocks. Les résultats pour les deux ensembles d'instances sont discutés.

Enfin, la Section 3 introduit un algorithme itératif qui résout des sous-problèmes issus de la formulation de l'IRP, ajoutant à chaque itération des solutions partielles pour accélérer la convergence. Les résultats expérimentaux sont également analysés.

4.8 L'*Inventory Routing Problem*

Même si plusieurs variations de l'IRP existent, certaines hypothèses sur le problème demeurent constantes à travers chaque variante. Ces hypothèses sont fondamentales pour comprendre les bases du problème et comment les aspects de la gestion des transports et des stocks interagissent. Les plus importantes sont :

- **Le *split delivery* n'est pas autorisé.** Chaque client est visité une seule fois par période et au maximum autant de fois que le nombre de périodes de l'horizon de temps.
- **Livraison et demande.** La livraison a lieu uniquement au début d'une période, et la demande est déduite de manière linéaire sur l'intervalle de temps considéré.
- **Calcul des niveaux de stock.** Les niveaux de stock sont calculés à la fin de la période, une fois toutes les opérations terminées. Ces opérations incluent le réapprovisionnement (livraison d'une quantité de produits), la déduction des demandes et le niveau de stock de la période précédente immédiate.
- **Capacité des véhicules.** Les véhicules chargés des livraisons ont une capacité finie et ne peuvent quitter le dépôt que s'ils ont au moins un client à servir, c'est-à-dire avec au moins une unité de produit à bord.
- **Disponibilité des produits à livrer.** Le fournisseur dispose toujours de suffisamment de produits à chaque période pour satisfaire toutes les demandes des clients.
- **Niveaux de stock maximum et minimum.** Les clients ont des niveaux de stock minimum et maximum à respecter. En général, le niveau minimum est égal à zéro. Pour le fournisseur, seul le niveau minimum existe et est toujours égal à zéro.
- **Disruption de stock.** En aucun cas, une disruption de stock n'est autorisée.
- **Faisabilité des itinéraires.** Un itinéraire valide commence et se termine toujours au dépôt. Un sous-tour (un itinéraire ne commençant ni ne se terminant au dépôt) doit être éliminé par des contraintes spéciales. De plus, chaque route est attribuée à un seul véhicule.

4.9 Formulations mathématiques

4.9.1 l'IRP

L'IRP classique consiste à déterminer quand et combien de produits doivent être livrés à un ensemble de clients de manière à éviter toute rupture de stock tout en respectant les niveaux de stock des clients et la capacité de production du fournisseur. Pour cela, une flotte de véhicules homogènes est considérée à chaque période, et les itinéraires à effectuer doivent également être définis. Les demandes des clients et la capacité de production du fournisseur sont considérées comme déterministes. L'objectif est de définir une solution minimisant les coûts de transport et de gestion des stocks, tout en respectant un ensemble de contraintes relatives à la gestion des stocks, aux quantités à livrer, à l'évitement des sous-tours et à la capacité des véhicules.

Formellement, l'IRP et l'HIRP-BS sont définis sur un graphe $G = (\mathcal{N}', \mathcal{A})$, où $\mathcal{N}$ représente l'ensemble des n clients $\mathcal{N} = 1, ..., n$ et le nœud 0 représente le fournisseur, avec $\mathcal{N}' = 0 \cup \mathcal{N}$. Ainsi, $\mathcal{A} = (i, j) : i, j \in \mathcal{N}', i \neq j$ est l'ensemble des arcs. Un horizon temporel $\mathcal{T} = 1, ..., H$ avec H périodes est considéré, et donc $\mathcal{T}' = 0 \cup \mathcal{T}$. Une flotte homogène de m véhicules, chacun ayant une capacité B, est utilisée. La matrice des distances $C = (c_{i,j})0 \leq i, j \leq |\mathcal{N}'|$ indique le coût ci, j pour voyager de i à j et respecte les inégalités triangulaires. Dans le cas de l'HIRP-BS, cette distance ne respecte pas forcément les inégalités triangulaires.

Un niveau de stock initial s_i, $\forall i \in \mathcal{N}'$, est connu à l'avance pour les clients et le fournisseur à la période 0. Les coûts de stockage sont donnés par h_i^t, avec $i \in \mathcal{N}'$ et $t \in \mathcal{T}'$. Chaque client a une demande indépendante de la période d_i et des niveaux de stock maximum U_i autorisés par période $t \in \mathcal{T}$.

Un résumé concernant la fonction objectif et les contraintes est présenté ci-dessous pour l'IRP et l'HIRP-BS.

- **Fonction objectif.** La fonction objective vise à minimiser la somme des coûts de stockage et de routage. Les coûts de stockage concernent les produits stockés chez les clients et le fournisseur, tandis que les coûts de routage concernent la distance totale parcourue pour servir tous les clients sur l'ensemble des périodes. Dans le cas de l'HIRP-BS, les coûts incluent les coûts fixes et variables selon l'utilisation du véhicule et la distance parcourue.

- **Contraintes**

  - **Niveaux de stock.** Au départ (à la période 0), les variables des niveaux de stock du fournisseur et des clients, I_i^t, $\forall i \in \mathcal{N}'$ et $t \in \mathcal{T}'$, sont fixées à leur niveau de stock initial s_i. Le niveau de stock du fournisseur est calculé en prenant le niveau de stock de la période précédente I_0^{t-1}, $\forall t \in \mathcal{T}$, en ajoutant sa capacité de production $r^t, \forall t \in \mathcal{T}$, et en déduisant le montant total livré aux clients, soit $\sum_{i \in \mathcal{N}} q_i^t$, $\forall t \in \mathcal{T}$. Le niveau de stock pour chaque client $i \in \mathcal{N}$ et chaque période $t \in \mathcal{T}$ est calculé en prenant en compte le niveau de stock précédent I_i^{t-1}, la quantité de produits livrée par le fournisseur, et les demandes pour la période en cours.

  - **Quantités à livrer.** La quantité q_i^t, $\forall i \in \mathcal{N}, t \in \mathcal{T}$ à livrer doit respecter l'espace de stockage disponible chez le client, soit la différence entre son niveau de stock précédent et sa capacité de stockage maximale ($U_i - I_i^{t-1}$ $\forall i \in \mathcal{N}, t \in \mathcal{T}$).

Il est nécessaire que le client recevant des produits à une période donnée soit visité sur l'un des itinéraires de cette même période.

– **Flux/degré.** Aussi appelées contraintes de flux, elles garantissent que le flux entrant et sortant à chaque nœud est égal. Impose que le nombre m de véhicules disponibles soit respecté en considérant le premier arc $x_{0,i}^{k,t}$ $\forall i \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K}^t$ dans une route donnée. Assure que chaque client est visité au plus une fois par période, puisque les livraisons fractionnées ne sont pas autorisées.

– **Capacité des véhicules et élimination des sous-tours.** Pour éviter les sous-tours, des variables $a_{i,j}^{k,t}$, $\forall i, j \in \mathcal{A}, t \in \mathcal{T}, k \in \mathcal{K}^t$ sont introduites et servent de compteur croissant pour garantir qu'une route commence et se termine au nœud 0, qui représente le fournisseur. Toutes les variables $a_{i,0}^{k,t}$, $\forall i \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K}^t$, revenant au fournisseur, sont fixées à 0. Les contraintes de conservation de flux veillent à ce que la charge des véhicules soit correctement gérée lors de l'arrivée et du départ de chaque client à chaque période. La charge des véhicules le long de chaque itinéraire est limitée par la capacité des véhicules $B^{k,t}$.

4.10 Métaheuristique basée sur un algorithme Split

La matheuristique basée sur le Split intègre un algorithme Greedy Adaptive Search Procedure (GRASP) qui utilise le mécanisme Split à différentes étapes de son processus, en combinaison avec une Recherche Locale Évolutionnaire (ELS) et une phase de Post-Optimisation, incluant une recherche locale. L'algorithme est nommé *Split-Embedded Metaheuristic with a Post-optimization* (SEMPO). La métaheuristique GRASP est une méthode bien connue qui construit itérativement des solutions initiales et applique une procédure de recherche locale jusqu'à atteindre un critère d'arrêt prédéfini. De nombreux chercheurs ont appliqué cette méthode aux problèmes de VRP (Kontoravdis and Bard 1995, Villegas et al. 2011, Guemri et al. 2016) ainsi qu'aux algorithmes ELS (Duhamel et al. 2011, Zhang et al. 2015). Bien que GRASP et ELS aient été largement explorés pour les VRP, leur application spécifique à l'IRP a été moins étudiée, avec aucune publication récente identifiée.

Les étapes de la métaheuristique sont détaillées ci-dessous.

- **Étape 1 :** Une heuristique constructive en trois étapes qui fournit des solutions réalisables même pour des instances de grande taille :
  - Définition d'un tour géant multi-période contenant une séquence de clients à visiter et les quantités à livrer, favorisant une décomposition en routes de qualité raisonnable.
  - Évaluation du tour géant généré pour décomposer la séquence en routes assignés aux véhicules disponibles à chaque période, à l'aide de l'algorithme Split.
  - Amélioration de la solution obtenue par un mécanisme de recherche locale agissant sur la partie routage du problème via quatre voisinages, considérant les mouvements intra et inter-routes. La solution améliorée est convertie en tour géant pour relancer le processus si nécessaire.
- **Étape 2 :** Une procédure de Recherche Locale Evolutionnaire (ELS) pour améliorer la solution obtenue à l'étape précédente, par un nombre défini d'itérations sans amélioration. L'ELS comprend :

- Une phase de mutation pour diversifier en permutant les positions du tour géant entre celles appartenant à la même période.
- Une phase d'évaluation (Split) et un mécanisme de recherche locale similaires à ceux de l'heuristique constructive en trois étapes décrite ci-dessus.

- **Étape 3 :** Le Post-Optimisation vise à affiner la solution en améliorant la meilleure solution ELS obtenue lors de l'itération GRASP actuelle. Cette méthode repose sur la définition d'une "distance" par rapport à la meilleure solution ELS pour contraindre l'espace de recherche. En restreignant cet espace, cette approche réduit l'effort computationnel nécessaire pour résoudre un modèle MILP tout en se concentrant sur les zones avec une probabilité plus élevée d'amélioration. Pendant cette étape, les quantités de livraison et les configurations des itinéraires sont optimisées, en ajoutant des contraintes liées à la "distance" de la solution ELS et en visant des solutions avec des itinéraires restant "proches" de ceux identifiés lors de l'étape précédente.

4.10.1 Algorithme Split

L'algorithme Split pour le VRP se concentre sur la recherche du chemin le plus court dans un graphe. Pour la version capacité (CVRP), il ne cherche pas seulement le chemin le plus court, mais prend également en compte les contraintes de ressources dues au nombre limité de véhicules. Dans le cas de l'IRP, l'algorithme Split intègre également la gestion des stocks dans le processus de création des itinéraires. L'objectif principal de cet algorithme est de résoudre efficacement les défis combinés du routage et de la gestion des stocks.

L'originalité de l'approche introduite dans cette thèse est illustrée par deux cas particuliers : (i) anticiper une livraison d'un client assigné à une période t vers une période t' telle que $t' < t$ et que le client n'ait pas encore été traité dans une route en cours, et (ii) le client a déjà été traité dans la route en cours.

L'algorithme suit les étapes suivantes :

- **Étape 1 :** Définition d'un grand tour. Générer un tour géant multi-période contenant une séquence de clients avec des quantités prédéfinies à livrer et une estimation du dernier moment possible pour effectuer ces livraisons. Cette répartition est réalisée par une heuristique constructive qui détermine le "meilleur" moment où un client doit être placé, ainsi que la quantité et la période à lui assigner.

- **Étape 2 :** Découpage du tour géant. De gauche à droite, découper la séquence en tenant compte des contraintes du problème, notamment la capacité de stockage des clients, le niveau des stocks et la capacité des véhicules. Cela inclut une exploration des routes possibles en ajoutant des labels contenant des informations sur les routes candidates. Une limite est imposée pour réduire la complexité, car un label peut en générer plusieurs autres. Les meilleurs candidats sont ensuite triés par coût croissant. Une fois le dernier client de la séquence choisi, la recherche se termine.

- **Étape 3 :** Récupération du chemin critique. De droite à gauche, cette étape vise à récupérer le chemin critique correspondant à la solution à coût minimal obtenue après l'exploration du tour géant. En choisissant un label sur le dernier client de la séquence, on trouve une solution faisable en identifiant les clients à l'origine de la route, ainsi que le véhicule et la période assignés.

4.10.2 Résultats

Version classique de l'IRP

L'algorithme SEMPO proposé a été conçu pour résoudre le HIRP-BS. Cette section présente les résultats obtenus par l'approche SEMPO lorsqu'elle est appliquée à la version classique de l'IRP, une variante plus simple du HIRP-BS. Bien que la méthode proposée donne des résultats satisfaisants pour les instances IRP, il est attendu que des méthodes spécialisées disponibles dans la littérature pour l'IRP obtiennent de meilleures performances.

L'algorithme SEMPO produit des solutions de haute qualité pour les instances classiques de l'IRP, avec un écart moyen de 2% par rapport aux meilleures solutions connues dans la littérature, tout en nécessitant beaucoup moins de temps de calcul (une moyenne de 11 secondes contre 33 minutes). Aucune solution trouvée par la formulation proposée, résolue avec CPLEX, n'a été prouvée optimale dans la limite d'une heure. De plus, pour deux petites instances, le modèle MILP n'a pas fourni de solution faisable, alors que l'algorithme SEMPO a généré des solutions pour toutes les instances. SEMPO a surpassé la formulation MILP en fournissant de meilleures solutions pour quatre instances en environ la moitié du temps. Pour les autres instances, les coûts des solutions étaient très proches des meilleures solutions obtenues par les deux méthodes.

Nouvelle variante HIRP-BS

Pour les instances de petite taille, l'algorithme SEMPO a atteint un écart moyen de 0,31% avec un temps de calcul moyen de 850 secondes pour trouver la meilleure solution, soit quatre fois plus rapide que le temps de 3519 secondes de la formulation proposée. De plus, la faible déviation standard des coûts parmi les 10 meilleures solutions confirme la stabilité de la métaheuristique. Le coût moyen des meilleures solutions fournies par SEMPO était de 6,67% supérieur au coût de la relaxation linéaire.

L'écart moyen entre le modèle MILP et SEMPO est d'environ 1%, avec une faible déviation standard. Pour trois instances, cet écart est négatif. En ce qui concerne la relaxation linéaire, l'écart est d'environ 7%, avec également une faible déviation standard.

Pour les instances de taille moyenne, l'écart moyen est de 8,53%, et SEMPO trouve la meilleure solution en moins de 20 minutes en moyenne (921 secondes).Ces résultats sont considérés de haute qualité compte tenu de la complexité du HIRP-BS, qui comprend des caractéristiques telles qu'une flotte hétérogène de véhicules, des coûts et demandes variables, et des tailles de lots par client. Cela se reflète dans le fait que, même pour des instances avec 19 clients et un horizon de 14 périodes, CPLEX n'a pas trouvé de solutions faisables dans le temps imparti d'une heure.

Enfin, bien que la phase de post-optimisation n'ait pas été exécutée pour les instances de grande taille, l'écart entre le coût des solutions fournies par SEMPO et celui de la relaxation linéaire n'est pas significativement plus grand comparé aux petites et moyennes instances, montrant ainsi l'efficacité de SEMPO en termes de temps d'exécution et de qualité des solutions.

4.11 Méthode itérative sur périodes

Soit $\mathcal{P}$ le problème IRP, et soit $\mathcal{P}^t$, pour tout $t \in \mathcal{T}$, le $t^{\text{ème}}$ sous-problème parmi les $|\mathcal{T}|$ sous-problèmes possibles. La résolution de $\mathcal{P}^t$ commence à partir des valeurs préalablement déterminées pour les compositions de routes, exprimées par les variables $x^t_{i,j}$ des périodes 1 à $t-1$, puis consiste à résoudre le sous-problème actuel $\mathcal{P}^t$, en permettant une certaine liberté pour les variables x lors de l'exécution de l'algorithme pour $\mathcal{P}^t$.

Au fur et à mesure que l'algorithme progresse à travers les périodes, la taille et la complexité des sous-problèmes augmentent. Plus précisément, les dimensions des sous-problèmes croissent à chaque itération successive, ce qui signifie que $|\mathcal{P}^1| < |\mathcal{P}^2| < \cdots < |\mathcal{P}^{\mathcal{T}-1}| < |\mathcal{P}^{\mathcal{T}}|$. Cette expansion progressive des sous-problèmes reflète l'accumulation d'informations et l'augmentation des décisions à prendre à mesure que l'algorithme approche de la période finale.

L'idée sous-jacente est d'explorer partiellement le voisinage associé à chaque période temporelle lors de chaque itération. Cette exploration est guidée par les variables qui représentent chaque période dans l'horizon temporel, permettant au solveur de suivre implicitement une séquence logique alignée avec l'ordre chronologique des périodes. Cela aide également le solveur à mieux comprendre la structure du problème en termes de périodes, ce qui peut ne pas être évident lorsqu'on considère toutes les variables séparément. Une telle approche est couramment utilisée dans les algorithmes de programmation par contraintes, comme mentionné dans des travaux précédents Bourreau et al. (2019, 2020).

Une fois que la résolution de $\mathcal{P}^t$ commence, en partant des valeurs connues des variables x jusqu'au période $t-1$, les valeurs correspondantes des q peuvent être déterminées plus facilement en fonction de l'utilisation ou non d'un arc dans un itinéraire. Cependant, lors de la transition du sous-problème $\mathcal{P}^{t-1}$ vers $\mathcal{P}^t$, une valeur initiale pour les variables q est fixée sans permettre de degré de liberté, ce qui guide le processus de recherche.

Le paramètre Δ indique le pourcentage de modifications autorisées dans chaque sous-problème $\mathcal{P}^t$ en fonction des arcs activés précédemment. Plus précisément, lorsque $\Delta = 0$, aucune modification des arcs précédemment activés n'est autorisée, tandis que lorsque $\Delta = 100$, tous les arcs activés précédemment peuvent potentiellement être modifiés.

4.11.1 Résultats

En moyenne, la méthode itérative est dix fois plus rapide que la résolution directe, avec un écart moyen de 9% sur la qualité des solutions. Cependant, une analyse plus détaillée, instance par instance, révèle une variabilité significative des performances. Plus précisément, cinq instances affichent une accélération du temps de calcul par un facteur dix. Parmi celles-ci, l'instance abs1n25.dat présente un écart de 19%, soulignant ainsi un compromis entre rapidité et qualité de la solution dans certains cas. L'instance la moins favorable est low abs1n5.dat, avec un horizon temporel de trois périodes, où un écart de 22% et un ratio de temps de calcul de 1 sont observés. En revanche, l'instance la plus favorable est high abs1n20.dat, également avec un horizon de trois périodes, montrant un écart de seulement 4% et un ratio de temps de calcul de 39.

Au cours de la première période, pour la majorité des instances analysées, aucune route n'est planifiée car les clients disposent de niveaux de stock suffisants, ce qui signifie qu'aucune livraison de produits n'est nécessaire. Par conséquent, il n'y a pas de différence entre les méthodes à ce stade. En avançant vers la deuxième période, toutes les valeurs de *gap* sont zéro. Cela s'explique par le fait que l'information transmise de la première période facilite grandement la résolution du sous-problème, garantissant que les deux méthodes aboutissent à la même solution.

Cependant, à partir de la troisième période et au-delà, la complexité du problème augmente considérablement. Dès le *gap* pour la période 3, la dimension du problème peut s'étendre de manière significative, rendant plus difficile l'obtention de solutions optimales. C'est à ce stade que la méthode itérative commence à se différencier en termes de performance de l'approche classique, car le problème devient de plus en plus complexe à résoudre avec chaque période successive.

Indépendamment du type ou de la taille de l'instance, une tendance est observée : à mesure que la valeur de Δ augmente, il y a une réduction notable du gap. Cette tendance s'explique par le fait qu'une valeur plus élevée de Δ permet au sous-problème d'avoir plus de flexibilité pour ajuster les arcs, augmentant ainsi les chances de trouver de meilleures solutions. Cependant, cet avantage a un coût : le temps de calcul augmente car l'espace de recherche s'étend avec des valeurs de Δ plus grandes. Malgré cela, même dans les scénarios les plus défavorables en termes de temps, l'approche itérative dépasse généralement la méthode classique en moyenne, mettant en évidence son efficacité à équilibrer la qualité des solutions et l'effort computationnel.

Conclusion

Cette thèse a traité du problème intégré de gestion des stocks et de routage appelé Inventory Routing Problem (IRP), ainsi que d'une nouvelle extension nommée Heterogeneous Inventory Routing Problem with Batch Size (HIRP-BS). Les deux posent des défis importants en termes d'opérations logistiques dans la chaîne logistique et doivent être coordonnés pour éviter les interruptions de stocks chez les clients et maintenir un niveau de service satisfaisant pour les clients comme pour le fournisseur. Pour ce faire, plusieurs approches ont été proposées. Une formulation mathématique, une matheuristique et un algorithme itératif ont été présentés.

Si les contributions majeures et l'originalité de cette thèse devaient être mises en avant, les aspects suivants pourraient être mentionnés :

- Introduction d'une variation de l'IRP (HIRP-BS) considérant une flotte de véhicules hétérogène dépendante des périodes, des demandes et des coûts de stockage, ainsi qu'une taille de lot par client
- Un nouvel ensemble d'instances pour aborder les caractéristiques du HIRP-BS
- Un algorithme Split capable de générer des solutions réalisables à partir d'un tour géant multi-périodes défini stratégiquement pour gérer les demandes des clients sur les périodes
- Une extension de la métaheuristique classique Grasp×ELS en incorporant une phase de post-optimisation utilisant la formulation linéaire du problème et un solveur commercial
- Un algorithme itératif de décomposition de domaine qui utilise les informations des itérations précédentes pour accélérer la convergence.

Comme pour tous les problèmes de recherche, il y a toujours place pour des travaux supplémentaires afin d'ajouter de nouvelles fonctionnalités et d'explorer d'autres caractéristiques de l'IRP, y compris de nouvelles contraintes et ensembles d'instances.

En parallèle de cette thèse, les algorithmes présentés ici peuvent fournir des bonnes perspectives pour le développement d'avantage d'algorithmes et pour leur adaptation afin d'incorporer de nouvelles fonctionnalités et de traiter d'autres problèmes de RO.

Appendix A

Instances examples

Go to the Table of Contents ↺
Go to the Instances section ↺

In this Appendix, examples of the classical IRP and the new HIRP-BS instances are presented. For each, a *How to read* and a file example provide details on how to read and how the files are structured, respectively.

A.1 Classical IRP instance example

These classical instances come from Archetti et al. (2007) and Archetti et al. (2012) for the single-vehicle and the multi-vehicle case, respectively. Note that for these two cases, the instances file remains the same: what changes is the readability of the vehicles capacity that is divides by the number of vehicles available and then rounded down for the nearest integer value.

In Subsection A.1.1, the details on how to read the file are given and then, in Subsection A.2.1, an example file is presented.

A.1.1 How to read

In order to better understand how these new instances files are structured, the following algorithm provide details on how to interpret them based on the initial problem data that can be found in Table 2.1.

$|\mathcal{N}|\ \ |\mathcal{T}|\ \ B$

Supplier data

$1\ \ s_0$
$X_1\ \ Y_1\ \ s_0\ \ r^t\ \ h_0^t$

Customers data

for each customer $i = 1$ to $|\mathcal{N}|$
$\quad$ *index* $X_i\ \ Y_i\ \ s_i\ \ U_i\ \ L_i\ \ d_i^t\ \ h_i^t$

A.1.2 Example file

File A.1: Literature instance `abs1n20`

21	6	1651					
1	113.0	95.0	2743		1101		.30
2	266.0	433.0	24	36	0	12	.23
3	257.0	469.0	91	182	0	91	.32
4	363.0	330.0	91	182	0	91	.33
5	158.0	453.0	45	90	0	45	.23
6	423.0	238.0	174	261	0	87	.18
7	363.0	368.0	78	117	0	39	.29
8	182.0	3.0	15	30	0	15	.42
9	332.0	420.0	55	110	0	55	.42
10	388.0	385.0	21	42	0	21	.24
11	188.0	69.0	184	276	0	92	.43
12	374.0	148.0	102	153	0	51	.18
13	296.0	322.0	20	40	0	20	.22
14	332.0	204.0	47	94	0	47	.24
15	432.0	250.0	72	144	0	72	.31
16	488.0	307.0	23	46	0	23	.22
17	46.0	351.0	86	129	0	43	.38
18	302.0	139.0	142	213	0	71	.13
19	23.0	126.0	176	264	0	88	.43
20	22.0	79.0	80	160	0	80	.37
21	81.0	442.0	116	174	0	58	.17

The instance `abs1n20` accounts for 20 customers as shown by the number 21 which also includes the supplier; 6 periods and the vehicle capacity of 1651 units of products per period. In the first line, index 1 represents the supplier followed by its `x` and `y` Euclidian coordinates, respectively, its initial inventory level of 2743 (fictitious period 0), its production capacity per period of 1101 units and its inventory holding costs of 0.30. The following lines represent each of the five customers and the information provided includes, in that order, the customer index 2 (shifted by 1 since the first stands for the supplier), the `x` and `y` Euclidian coordinates 266 and 433, respectively, the initial inventory level of 24, the maximum and minimum inventory level of 36 and 0 authorized, the static demand of 12 and an inventory holding cost of 0.23. The next four lines gives provide the same information for the remaining four customers.

A.2 New benchmark IRP instance example

A.2.1 How to read

In order to better understand how these new instances files are structured, the following algorithm provide details on how to interpret them based on the initial problem data that can be found in Table 2.2. Three main parts are considered: the vehicles, supplier and customers data. The first contains, for each period, information about the vehicles available, in which quantity, their capacity and costs involved. The second provides the supplier initial inventory level, inventory cost and production capacity. The third related to the customers concerns their initial inventory level, maximum inventory level allowed, batch size, inventory costs, demands as well as the distances among them considering a complete graph.

$|\mathcal{N}|$ $|\mathcal{T}|$

Vehicles data

for each period $t \in \mathcal{T}$
  t $|\mathcal{K}^t|$
  for each type $k \in \mathcal{K}^t$ of vehicle available
    k $m^{k,t}$ $B^{k,t}$ $f^{k,t}$ $v^{k,t}$

Supplier data

0 s_0
for each period $t \in \mathcal{T}$
  h_0^t r^t

Customers data

for each customer $i \in \mathcal{N}$
  i s_i U_i ℓ_i h_i^t d_i^t
for each $i \in \mathcal{N}'$
  for each $j \in \mathcal{N}'$
    $c_{i,j}$

A.2.2 Example file

File A.2: New instance s_19_7_1

```
19 7
1 3
1 3 227 126 0.85
2 1 221 89 1.30
3 1 197 115 2.97
2 3
1 2 246 126 0.63
2 3 218 122 1.72
3 1 171 91 1.44
3 2
1 4 282 92 2.69
2 1 276 144 0.73
4 3
1 4 269 139 2.14
2 1 244 59 0.97
3 1 191 127 1.22
5 1
1 3 359 100 0.72
6 2
1 2 217 75 2.56
2 4 194 65 2.56
7 4
1 4 129 90 1.64
2 1 112 90 0.95
3 4 95 111 1.44
4 2 92 63 1.95
0 773 0.34 1664 0.44 1636 0.18 2039 0.38 1432 0.34 1643 0.49 2052 0.17 1353
1 173 0 173 13 0.1 19 0.2 95 0.5 27 0.4 60 0.1 47 0.2 91 0.4 10
2 60 0 145 14 0.5 38 0.4 42 0.4 33 0.2 24 0.3 14 0.4 10 0.2 85
3 114 0 161 17 0.3 30 0.2 36 0.2 88 0.4 10 0.3 40 0.2 89 0.5 33
4 115 0 115 6 0.3 70 0.5 96 0.4 51 0.3 87 0.4 32 0.2 93 0.3 29
5 57 0 156 16 0.2 81 0.2 11 0.1 84 0.3 61 0.4 82 0.5 90 0.4 40
6 42 0 166 11 0.4 77 0.1 80 0.2 43 0.4 60 0.2 28 0.3 22 0.2 84
7 43 0 97 13 0.2 15 0.4 72 0.4 47 0.1 64 0.3 82 0.3 14 0.3 37
8 147 0 158 4 0.3 19 0.1 11 0.3 83 0.1 51 0.2 73 0.1 79 0.4 22
9 84 0 149 3 0.3 74 0.4 40 0.1 24 0.2 77 0.2 55 0.2 42 0.3 28
10 46 0 98 2 0.2 87 0.3 20 0.3 29 0.5 81 0.5 47 0.3 43 0.1 11
11 91 0 118 14 0.2 27 0.3 70 0.4 73 0.1 37 0.4 74 0.1 63 0.5 80
12 99 0 185 5 0.1 33 0.5 78 0.3 18 0.4 45 0.3 23 0.4 99 0.4 90
13 70 0 122 6 0.5 40 0.2 94 0.3 98 0.3 89 0.2 74 0.3 72 0.1 28
14 25 0 76 7 0.4 51 0.5 69 0.1 43 0.5 43 0.5 69 0.3 42 0.5 43
15 38 0 91 4 0.3 56 0.3 16 0.1 84 0.1 70 0.2 21 0.5 23 0.3 12
16 129 0 157 6 0.3 95 0.3 40 0.2 76 0.3 86 0.2 18 0.2 46 0.5 74
17 103 0 123 12 0.3 79 0.3 19 0.3 18 0.3 77 0.3 23 0.2 86 0.4 87
18 56 0 102 2 0.1 12 0.3 34 0.2 47 0.5 47 0.3 57 0.3 69 0.4 94
19 54 0 156 7 0.1 52 0.2 91 0.1 60 0.3 13 0.1 11 0.4 46 0.2 93
0       1238    2398    2183    3214    2930    3709    2627    2677    4229    4627    7891    6341    5047    8006    8159    6765    5946    6358    7358
1238    0       1494    2632    3662    3379    4158    2981    2021    2146    3635    8387    6790    5496    8455    8608    4640    3305    4275    6364
2398    1494    0       1601    2796    3238    5164    5015    4078    2793    3192    7260    5882    5355    9456    9609    8215    4510    4922    5921
2183    2632    1601    0       1891    2612    4538    4389    5695    4410    2728    8321    4734    4728    8829    8982    7588    6127    6539    10465
3214    3662    2796    1891    0       2215    4969    5555    5706    5101    3516    8540    2461    3515    5884    10149   8755    6818    7230    10685
2930    3379    3238    2612    2215    0       3473    3747    4835    5363    5762    10464   4635    2941    3534    8340    6946    7080    7492    12609
3709    4158    5164    4538    4969    3473    0       2414    4284    7443    7841    11105   6156    3717    2665    5409    5614    6987    9572    10572
2627    2981    5015    4389    5555    3747    2414    0       2529    4682    7534    10798   9248    5936    5609    5163    3257    5280    6999    10265
```

2677	2021	4078	5695	5706	4835	4284	2529	0	2262	4024	8782	8279	6985	8671	12308	3795	2634	4353	6759
4229	2146	2793	4410	5101	5363	7443	4682	2262	0	2963	7715	7088	6674	10782	13585	6304	3788	2695	5692
4627	3635	3192	2728	3516	5762	7841	7534	4024	2963	0	4835	6483	9225	16701	14517	7359	4843	4670	2912
7891	8387	7260	8321	8540	10464	11105	10798	8782	7715	4835	0	5588	9427	13244	16886	16800	13183	8395	5501
6341	6790	5882	4734	2461	4635	6156	9248	8279	7088	6483	5588	0	3283	6031	13975	11825	16986	12198	9304
5047	5496	5355	4728	3515	2941	3717	5936	6985	6674	9225	9427	3283	0	3751	9949	14877	8610	15703	12809
8006	8455	9456	8829	5884	3534	2665	5609	8671	10782	16701	13244	6031	3751	0	4242	7261	15593	20098	17203
8159	8608	9609	8982	10149	8340	5409	5163	12308	13585	14517	16886	13975	9949	4242	0	5068	9132	13855	17429
6765	4640	8215	7588	8755	6946	5614	3257	3795	6304	7359	16800	11825	14877	7261	5068	0	5031	10025	13600
5946	3305	4510	6127	6818	7080	6987	5280	2634	3788	4843	13183	16986	8610	15593	9132	5031	0	3757	10358
6358	4275	4922	6539	7230	7492	9572	6999	4353	2695	4670	8395	12198	15703	20098	13855	10025	3757	0	5639
7358	6364	5921	10465	10685	12609	10572	10265	6759	5692	2912	5501	9304	12809	17203	17429	13600	10358	5639	0

Appendix B

IRP extended results

Go to the Table of Contents
Go to the IRP results section

In this Appendix, the tables present the results for the IRP classical instances set from Archetti et al. (2007) for single-vehicle case and Archetti et al. (2012); Coelho et al. (2012a) for the multi-vehicle case per period. The instances were solved through the linear flow formulation presented in Section 2.2 from Chapter 2.

B.1 Tables description

For each table, the results are provided for both high and low inventory holding costs. Column *abs* provides the instance type from $\{1, 2, ..., 5\}$, $|\mathcal{N}|$ to the number of customers, z_{LB} and z_{UB} for the lower and upper bounds, respectively, $t(s)$ to the time in seconds (1-hour limit for execution) and *gap* the gap between z_{LB} and z_{UB}.

As a reminder, these classical IRP instances are summarized in Table B.1.

Table B.1: Literature instances characteristics

Number of instances			$\mathcal{N}$	$\|\mathcal{T}\|$	h_i	h_0
160	100	50	{5, 10, ..., 50}	3	[0.01; 0.05]	0.03
		50	{5, 10, ..., 50}	3	[0.1; 0.5]	0.3
	60	30	{5, 10, ..., 30}	6	[0.01; 0.05]	0.03
		30	{5, 10, ..., 30}	6	[0.1; 0.5]	0.3

For the following, the tables are summarized below.

- **Table B.2.** 3 periods, 1 vehicle
- **Table B.3.** 3 periods, 2 vehicles
- **Table B.4.** 3 periods, 3 vehicles
- **Table B.5.** 3 periods, 4 vehicles
- **Table B.6.** 3 periods, 5 vehicles
- **Table B.7.** 6 periods, 1 vehicle
- **Table B.8.** 6 periods, 2 vehicles
- **Table B.9.** 6 periods, 3 vehicles
- **Table B.10.** 6 periods, 4 vehicles
- **Table B.11.** 6 periods, 5 vehicles

B.1.1 3-period instances

Table B.2: IRP results for 3 periods, 1 vehicle

		High inventory costs				Low inventory costs			
abs	$\|\mathcal{N}\|$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$
1	5	2108.34	2108.34	0.09	0	1235.92	1235.92	0.09	0
2	5	1767.06	1767.06	0.13	0	988.66	988.66	0.08	0
3	5	2973.00	2973.00	0.07	0	1758.02	1758.02	0.07	0
4	5	1981.04	1981.04	0.09	0	1397.20	1397.29	0.21	0
5	5	2170.04	2170.04	0.07	0	999.36	999.42	0.10	0
1	10	4510.61	4510.61	0.36	0	1743.07	1743.07	0.39	0
2	10	4504.61	4504.61	0.39	0	2229.25	2229.25	0.79	0
3	10	4031.40	4031.40	0.30	0	1871.14	1871.14	0.32	0
4	10	3933.46	3933.46	0.17	0	1773.00	1773.00	0.20	0
5	10	4709.35	4709.79	1.09	0	1938.18	1938.18	0.42	0
1	15	5589.20	5589.70	2.09	0	2131.00	2131.04	1.85	0
2	15	5443.02	5443.34	3.48	0	2131.37	2131.58	2.42	0
3	15	6300.27	6300.86	1.44	0	2463.68	2463.68	1.27	0
4	15	4977.16	4977.58	6.42	0	2151.94	2151.94	4.14	0
5	15	4867.38	4867.53	2.33	0	2160.42	2160.59	2.94	0
1	20	6858.74	6859.02	2.36	0	2267.32	2267.32	0.79	0
2	20	7087.24	7087.74	17.26	0	2497.69	2497.90	7.03	0
3	20	7354.53	7354.68	5.95	0	2590.48	2590.48	4.87	0
4	20	6952.10	6952.79	38.72	0	3122.03	3122.31	62.74	0
5	20	7874.26	7874.26	8.23	0	2849.90	2849.90	6.53	0
1	25	8227.18	8227.86	13.86	0	2840.92	2840.92	15.64	0
2	25	8765.32	8765.72	18.32	0	3014.56	3014.56	13.26	0
3	25	9382.11	9382.42	19.82	0	3050.40	3050.40	13.52	0
4	25	8452.09	8452.93	20.33	0	3078.45	3078.67	18.97	0
5	25	10080.50	10081.40	19.38	0	2954.71	2954.96	13.62	0
1	30	12066.00	12066.90	29.74	0	3427.49	3427.78	21.43	0
2	30	10940.30	10941.30	24.66	0	3328.79	3328.94	15.62	0
3	30	12121.20	12122.40	39.43	0	3471.78	3471.86	20.69	0
4	30	9686.13	9687.10	124.32	0	3321.16	3321.48	107.89	0
5	30	9773.08	9773.90	34.84	0	2914.41	2914.60	10.87	0
1	35	11659.60	11659.90	61.08	0	3315.11	3315.26	21.91	0
2	35	10465.80	10466.80	69.97	0	3229.22	3229.34	24.89	0
3	35	13775.50	13776.50	37.61	0	3811.40	3811.78	24.00	0
4	35	10306.40	10307.40	37.33	0	3346.12	3346.12	24.66	0
5	35	10846.90	10847.80	25.94	0	3541.39	3541.71	42.93	0
1	40	13363.60	13364.90	110.81	0	3874.62	3874.62	35.91	0

Continued on next page

Table B.2 – IRP results for 3 periods, 1 vehicle (continued)

		High inventory costs				Low inventory costs			
abs	$\|\mathcal{N}\|$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$
2	40	11316.70	11317.80	395.22	0	3701.91	3702.14	62.59	0
3	40	13598.20	13598.90	71.09	0	3534.80	3534.80	40.83	0
4	40	11353.40	11353.40	51.00	0	3575.46	3575.46	26.25	0
5	40	13069.70	13070.20	78.71	0	3831.71	3832.09	327.50	0
2	45	13142.20	13142.20	78.55	0	3950.86	3950.86	69.74	0
1	45	14177.70	14179.10	326.00	0	3702.72	3702.72	55.95	0
3	45	14842.30	14843.60	80.22	0	3967.77	3968.04	176.17	0
4	45	13573.30	13574.50	146.72	0	3998.21	3998.26	41.48	0
5	45	13585.90	13587.30	877.10	0	3717.17	3717.54	431.08	0
1	50	14575.90	14577.30	219.14	0	4046.91	4047.18	97.06	0
2	50	15000.20	15001.60	420.65	0	4512.79	4512.96	73.16	0
3	50	15278.00	15279.50	272.29	0	4451.00	4451.44	62.17	0
4	50	16515.40	16517.00	139.37	0	4405.84	4405.84	85.27	0
5	50	15677.10	15678.70	187.85	0	4218.10	4218.37	134.12	0

Table B.3: IRP results for 3 periods, 2 vehicles

		High inventory costs				Low inventory costs			
abs	$\|\mathcal{N}\|$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$
1	5	2265.12	2265.21	0.13	0	1396.33	1396.33	0.31	0
2	5	1969.31	1969.31	0.06	0	1177.49	1177.49	0.28	0
3	5	3653.00	3653.00	0.32	0	2437.93	2438.02	0.29	0
4	5	2301.04	2301.04	0.15	0	1717.29	1717.29	0.34	0
5	5	2372.36	2372.36	0.08	0	1220.21	1220.21	0.15	0
1	10	5031.58	5032.05	2.43	0	2263.10	2263.19	3.29	0
2	10	5080.45	5080.54	0.48	0	2809.71	2809.86	0.85	0
3	10	4371.55	4371.94	0.89	0	2220.46	2220.46	1.15	0
4	10	4642.81	4643.24	5.34	0	2481.89	2482.06	4.38	0
5	10	4930.79	4930.79	1.16	0	2159.04	2159.18	0.81	0
1	15	5754.99	5755.54	3.49	0	2296.92	2297.02	5.99	0
2	15	5852.79	5853.37	20.31	0	2553.85	2554.10	36.26	0
3	15	6646.05	6646.62	7.20	0	2799.80	2800.00	7.72	0
4	15	5337.83	5338.36	25.68	0	2513.45	2513.69	50.63	0
5	15	5317.01	5317.53	9.25	0	2610.43	2610.59	9.57	0
1	20	7508.01	7508.76	413.89	0	2917.02	2917.30	274.99	0
2	20	7254.26	7254.98	18.26	0	2664.98	2664.98	14.72	0
3	20	7591.45	7592.14	10.70	0	2818.40	2818.62	12.55	0
4	20	7379.85	7380.59	1170.10	0	3416.84	3417.18	46.33	0
5	20	8440.71	8441.54	23.95	0	3560.26	3560.62	1745.83	0
1	25	8520.71	8521.52	99.07	0	3133.01	3133.28	74.06	0
2	25	9173.89	9258.10	3600.00	1	3471.47	3471.82	243.86	0
3	25	9803.96	9804.92	142.76	0	3247.29	3247.61	132.56	0
4	25	8630.47	8631.33	61.94	0	3506.35	3506.70	60.96	0
5	25	10632.00	10633.10	55.75	0	3439.51	3501.14	3600.00	2
1	30	12448.60	12449.90	323.72	0	3803.40	3803.78	495.45	0
2	30	11257.00	11258.10	177.01	0	3644.85	3645.20	89.96	0
3	30	12275.90	12277.10	35.18	0	3615.84	3616.18	39.50	0
4	30	9837.02	9963.98	3600.00	1	3215.54	3215.86	1893.11	0
5	30	10062.50	10063.50	1431.12	0	3434.56	3609.98	3600.00	5
1	35	11921.10	11922.30	256.44	0	3593.29	3593.64	71.83	0
2	35	10764.60	10765.60	2988.85	0	3441.00	3441.32	39.54	0
3	35	14146.10	14147.50	177.95	0	4183.64	4184.06	117.61	0
4	35	10521.10	10522.10	43.10	0	3597.54	3597.90	292.38	0
5	35	11125.90	11127.00	47.70	0	3854.22	3854.60	1882.39	0
1	40	13659.00	13660.40	460.08	0	3736.41	3736.78	429.65	0
2	40	11554.40	11689.00	3600.00	1	4105.17	4105.58	510.79	0

Continued on next page

Table B.3 – IRP results for 3 periods, 2 vehicles (continued)

		High inventory costs				Low inventory costs			
abs	$\lvert\mathcal{N}\rvert$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$
3	40	13828.70	13830.10	556.63	0	3992.06	3992.46	763.68	0
4	40	11589.30	11590.40	1270.95	0	3994.42	4182.07	3600.00	4
5	40	13336.30	13413.50	3600.00	1	3868.22	3895.46	3600.00	1
2	45	14306.00	14307.30	446.07	0	4077.52	4077.86	82.93	0
1	45	13548.30	13651.00	3600.00	1	4121.31	4121.72	537.37	0
3	45	15006.70	15008.20	549.19	0	4068.92	4183.68	3600.00	3
4	45	13938.50	14048.30	3600.00	1	4383.08	4481.60	3600.00	2
5	45	13700.20	13792.40	3600.00	1	3829.70	3912.18	3600.00	2
1	50	14989.10	15136.40	3600.00	1	4662.37	4662.84	750.80	0
2	50	15341.40	15342.90	1769.40	0	4842.44	4842.92	1237.56	0
3	50	15488.50	15520.90	3600.00	0	4491.21	4614.16	3600.00	3
4	50	16773.40	16775.10	781.46	0	4667.89	4705.74	3600.00	1
5	50	15916.60	16123.60	3600.00	1	4497.82	4586.49	3600.00	2

Table B.4: IRP results for 3 periods, 3 vehicles

		High inventory costs				Low inventory costs			
abs	$\lvert\mathcal{N}\rvert$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$
1	5	2298.59	2298.59	0.12	0	1430.49	1430.49	0.19	0
2	5	2369.91	2369.93	0.32	0	1582.62	1582.69	0.42	0
3	5	4190.92	4191.26	0.13	0	2997.44	2997.44	0.32	0
4	5	2844.86	2844.94	0.18	0	2262.16	2262.16	0.18	0
5	5	2663.46	2663.66	0.23	0	1513.63	1513.76	0.43	0
1	10	5505.43	5505.85	3.69	0	2732.59	2732.59	5.23	0
2	10	5742.67	5743.20	3.77	0	3469.93	3470.14	5.02	0
3	10	4807.62	4808.06	4.61	0	2649.06	2649.26	3.66	0
4	10	5334.76	5335.29	23.75	0	3183.05	3183.36	25.63	0
5	10	5223.94	5224.45	2.71	0	2460.97	2461.11	2.93	0
1	15	6242.28	6242.90	39.46	0	2757.56	2757.82	4.55	0
2	15	6070.72	6071.26	1.80	0	3072.52	3072.80	3.42	0
3	15	6925.19	6925.82	3.59	0	2783.49	2783.77	204.65	0
4	15	5704.59	5705.16	277.64	0	2886.03	2886.32	127.35	0
5	15	5966.65	5967.25	1903.31	0	3184.39	3260.59	3600.00	2
1	20	7497.81	7498.56	156.90	0	3064.78	3064.78	11.53	0
2	20	7839.98	7840.54	9.74	0	2908.09	2908.38	176.34	0
3	20	9148.25	9149.16	1420.90	0	4123.83	4124.24	2181.87	0
4	20	8009.89	8165.42	3600.00	2	3418.07	3605.72	3600.00	5
5	20	7810.77	7918.67	3600.00	1	3976.75	4088.85	3600.00	3
1	25	8799.71	8893.88	3600.00	1	3394.13	3503.30	3600.00	3
2	25	9581.14	9676.58	3600.00	1	3835.64	3916.72	3600.00	2
3	25	10315.40	10404.00	3600.00	1	3938.63	4068.58	3600.00	3
4	25	8968.99	9052.69	3600.00	1	3578.45	3659.03	3600.00	2
5	25	11225.00	11250.90	3600.00	0	4033.00	4120.26	3600.00	2
1	30	12486.70	12488.00	116.80	0	3820.31	3820.66	144.98	0
2	30	12778.00	12918.30	3600.00	1	4080.44	4251.64	3600.00	4
3	30	11535.80	11737.60	3600.00	2	3920.00	4102.29	3600.00	4
4	30	10155.10	10287.10	3600.00	1	3755.97	3958.58	3600.00	5
5	30	10321.70	10449.90	3600.00	1	3445.20	3588.60	3600.00	4
1	35	12231.30	12435.70	3600.00	2	3876.51	4051.94	3600.00	4
2	35	10969.60	11125.30	3600.00	1	3813.19	4031.04	3600.00	5
3	35	14643.60	14694.50	3600.00	0	4690.73	4725.86	3600.00	1
4	35	10913.70	11106.40	3600.00	2	4047.61	4221.73	3600.00	4
5	35	11415.90	11567.50	3600.00	1	3853.24	4080.60	3600.00	6
1	40	13963.00	14229.80	3600.00	2	4309.71	4310.14	2398.89	0
2	40	11802.30	12015.50	3600.00	2	4285.62	4544.89	3600.00	6

Continued on next page

Table B.4 – IRP results for 3 periods, 3 vehicles (continued)

		High inventory costs				Low inventory costs			
abs	$\|\mathcal{N}\|$	z_{LB}	z_{UB}	$t(s)$	*gap*(%)	z_{LB}	z_{UB}	$t(s)$	*gap*(%)
3	40	14038.10	14039.50	1994.81	0	4116.35	4435.10	3600.00	7
4	40	11790.70	11849.00	3600.00	0	4252.12	4545.00	3600.00	6
5	40	13608.20	13976.30	3600.00	3	3955.42	3994.38	3600.00	1
2	45	15168.40	15170.00	1159.82	0	4266.99	4267.42	828.43	0
1	45	14637.00	14771.00	3600.00	1	4397.17	4537.30	3600.00	3
3	45	13902.90	14295.20	3600.00	3	4413.16	4825.74	3600.00	9
4	45	13887.50	14026.60	3600.00	1	3976.21	4154.82	3600.00	4
5	45	14247.20	14561.00	3600.00	2	4695.31	4921.10	3600.00	5
1	50	15504.40	15945.90	3600.00	3	4943.69	5162.78	3600.00	4
2	50	15746.10	15974.60	3600.00	1	4963.91	5207.12	3600.00	5
3	50	17142.90	17654.40	3600.00	3	5000.49	5289.80	3600.00	5
4	50	16350.90	16628.80	3600.00	2	4882.15	5183.63	3600.00	6
5	50	15688.90	16129.10	3600.00	3	5150.29	5791.58	3600.00	11

Table B.5: IRP results for 3 periods, 4 vehicles

		High inventory costs				Low inventory costs			
abs	$\|\mathcal{N}\|$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$
1	5	2411.27	2411.27	0.07	0	1541.73	1541.73	0.19	0
2	5	2452.97	2453.18	0.20	0	1665.74	1665.74	0.25	0
3	5	4807.52	4807.52	0.36	0	3603.72	3603.74	0.57	0
4	5	3210.24	3210.24	0.09	0	2631.56	2631.58	0.35	0
5	5	2824.32	2824.50	0.14	0	1676.40	1676.40	0.18	0
1	10	6020.82	6021.09	4.35	0	3261.63	3261.94	16.51	0
2	10	6476.60	6477.24	12.00	0	4208.36	4208.78	16.91	0
3	10	5126.45	5126.96	5.00	0	2968.49	2968.74	4.18	0
4	10	5834.10	5834.68	60.06	0	2871.81	2872.09	8.31	0
5	10	5643.83	5644.39	14.28	0	3682.81	3683.17	53.84	0
1	15	6704.78	6705.45	136.89	0	3396.28	3396.62	151.73	0
2	15	6610.65	6611.31	272.24	0	3199.80	3200.12	108.00	0
3	15	6016.24	6016.84	103.82	0	3545.25	3545.60	106.89	0
4	15	6255.63	6256.24	72.87	0	3756.99	3757.36	810.89	0
5	15	7606.92	7607.68	610.04	0	3166.12	3166.44	2569.22	0
1	20	7710.14	7710.91	235.19	0	3127.74	3128.05	317.26	0
2	20	9781.57	9782.55	1244.10	0	4764.30	4764.78	2978.92	0
3	20	8418.11	8717.83	3600.00	3	3774.55	4148.00	3600.00	9
4	20	8337.58	8414.08	3600.00	1	3528.92	3645.48	3600.00	3
5	20	8433.67	8592.35	3600.00	2	4465.17	4773.73	3600.00	6
1	25	9044.89	9287.53	3600.00	3	3648.97	3949.69	3600.00	8
2	25	10012.30	10278.30	3600.00	3	4272.33	4523.59	3600.00	6
3	25	10853.30	11026.80	3600.00	2	4520.37	4687.62	3600.00	4
4	25	9260.16	9438.34	3600.00	2	3854.63	4069.25	3600.00	5
5	25	11732.20	11805.50	3600.00	1	4578.40	4672.78	3600.00	2
1	30	12821.40	12822.70	685.05	0	4141.63	4142.04	1106.72	0
2	30	13144.90	13405.40	3600.00	2	4448.95	4758.06	3600.00	6
3	30	11900.00	12027.90	3600.00	1	4297.32	4453.99	3600.00	4
4	30	10500.80	10843.30	3600.00	3	4135.48	4468.71	3600.00	7
5	30	10621.70	10891.60	3600.00	2	3752.67	3991.70	3600.00	6
1	35	12560.70	12698.30	3600.00	1	4241.71	4521.10	3600.00	6
2	35	11288.30	11674.60	3600.00	3	5094.63	5341.60	3600.00	5
3	35	15089.90	15417.40	3600.00	2	4358.48	4740.22	3600.00	8
4	35	11279.10	11676.30	3600.00	3	4196.36	4608.06	3600.00	9
5	35	11785.20	12060.80	3600.00	2	4191.85	4379.12	3600.00	4
1	40	14362.80	14747.20	3600.00	3	4662.35	5107.19	3600.00	9
2	40	12178.20	12545.90	3600.00	3	4483.45	4671.50	3600.00	4

Continued on next page

Table B.5 – IRP results for 3 periods, 4 vehicles (continued)

		High inventory costs				Low inventory costs			
abs	$\lvert\mathcal{N}\rvert$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$
3	40	14225.90	14398.90	3600.00	1	4693.73	5070.26	3600.00	7
4	40	12063.70	12403.70	3600.00	3	4219.15	4456.70	3600.00	5
5	40	13990.70	14233.20	3600.00	2	4474.40	4756.60	3600.00	6
2	45	14990.20	15258.40	3600.00	2	4744.77	5035.20	3600.00	6
1	45	14408.10	14815.30	3600.00	3	4921.01	5341.00	3600.00	8
3	45	15341.80	15451.70	3600.00	1	4392.69	4514.50	3600.00	3
4	45	14095.90	14305.10	3600.00	1	4197.00	4426.96	3600.00	5
5	45	14711.70	15030.00	3600.00	2	5148.43	5741.86	3600.00	10
1	50	17672.60	18175.10	3600.00	3	5545.82	6057.92	3600.00	8
2	50	16141.00	16327.80	3600.00	1	5299.55	5624.07	3600.00	6
3	50	16054.60	16398.00	3600.00	2	5522.84	6164.42	3600.00	10
4	50	16172.80	16740.30	3600.00	3	5660.40	6406.86	3600.00	12
5	50	16844.60	17422.10	3600.00	3	5365.54	5905.22	3600.00	9

Table B.6: IRP results for 3 periods, 5 vehicles

		High inventory costs				Low inventory costs			
abs	$\lvert\mathcal{N}\rvert$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$
1	5	2577.40	2577.54	0.17	0	1710.34	1710.34	0.12	0
2	5	2806.05	2806.27	0.68	0	2019.45	2019.45	0.24	0
3	5	5115.64	5115.64	0.21	0	3918.88	3918.88	0.72	0
4	5	3899.39	3899.72	0.26	0	3318.97	3318.97	0.20	0
5	5	3166.28	3166.48	0.45	0	2008.29	2008.49	0.65	0
1	10	6495.31	6495.89	22.57	0	3728.35	3728.71	8.92	0
2	10	6959.75	6960.44	9.32	0	4680.83	4681.29	10.62	0
3	10	5555.79	5556.34	7.47	0	4156.55	4156.95	19.67	0
4	10	6307.90	6308.53	22.13	0	2993.77	2993.89	3.47	0
5	10	5770.91	5771.47	4.37	0	3400.76	3401.10	143.44	0
1	15	7022.97	7023.67	247.45	0	3580.45	3580.81	334.69	0
2	15	6386.96	6387.60	174.31	0	3572.12	3572.48	616.44	0
3	15	7976.46	7977.26	1582.22	0	3889.04	3889.43	2429.97	0
4	15	7193.58	7194.30	1935.98	0	4113.94	4128.41	3600.00	0
5	15	6836.51	6963.15	3600.00	2	4116.01	4260.88	3600.00	3
1	20	7909.71	7910.50	151.53	0	3343.78	3344.11	873.96	0
2	20	8964.31	8965.21	1008.92	0	4382.53	4404.77	3600.00	1
3	20	8620.94	8787.40	3600.00	2	3912.45	4016.48	3600.00	3
4	20	8918.73	9047.34	3600.00	1	5089.54	5219.21	3600.00	2
5	20	10375.00	10603.90	3600.00	2	5386.49	5506.02	3600.00	2
1	25	9363.20	9473.21	3600.00	1	4062.50	4095.20	3600.00	1
2	25	10586.00	10757.70	3600.00	2	4825.08	5009.76	3600.00	4
3	25	11477.10	11576.30	3600.00	1	5092.39	5229.32	3600.00	3
4	25	9552.42	9719.33	3600.00	2	4182.76	4378.75	3600.00	4
5	25	12262.70	12454.00	3600.00	2	5119.61	5315.02	3600.00	4
1	30	12266.20	12498.50	3600.00	2	4661.69	4879.36	3600.00	4
2	30	13040.70	13128.10	3600.00	1	4335.98	4458.84	3600.00	3
3	30	13570.70	14086.60	3600.00	4	4587.45	4959.78	3600.00	8
4	30	10936.70	11234.30	3600.00	3	4988.86	5411.16	3600.00	8
5	30	10987.00	11234.30	3600.00	2	4127.11	4433.58	3600.00	7
1	35	12909.90	13105.20	3600.00	1	4738.07	5027.59	3600.00	6
2	35	11668.30	12059.90	3600.00	3	4640.60	4864.66	3600.00	5
3	35	11746.50	12036.90	3600.00	2	4640.73	4845.42	3600.00	4
4	35	12172.60	12388.70	3600.00	2	4550.97	4764.86	3600.00	4
5	35	15620.50	15926.90	3600.00	2	5621.46	6036.76	3600.00	7
1	40	12592.40	12832.50	3600.00	2	4556.89	4781.78	3600.00	5
2	40	14471.80	14694.50	3600.00	2	4741.75	4977.14	3600.00	5

Continued on next page

Table B.6 – IRP results for 3 periods, 5 vehicles (continued)

		High inventory costs				Low inventory costs			
abs	$\lvert\mathcal{N}\rvert$	z_{LB}	z_{UB}	$t(s)$	*gap*(%)	z_{LB}	z_{UB}	$t(s)$	*gap*(%)
3	40	12423.10	12728.40	3600.00	2	5149.92	5354.91	3600.00	4
4	40	14791.80	15376.60	3600.00	4	5109.79	5647.94	3600.00	10
5	40	14383.30	14505.90	3600.00	1	4896.85	5006.26	3600.00	2
2	45	15569.20	15729.60	3600.00	1	4619.16	4831.70	3600.00	4
1	45	14334.40	14486.40	3600.00	1	4439.47	4686.29	3600.00	5
3	45	15221.20	15910.10	3600.00	4	5134.47	5520.42	3600.00	7
4	45	15375.20	15746.50	3600.00	2	5652.20	6310.31	3600.00	10
5	45	14976.10	15569.10	3600.00	4	5468.58	6141.00	3600.00	11
1	50	16654.30	17132.20	3600.00	3	5716.76	6151.10	3600.00	7
2	50	16789.40	17725.80	3600.00	5	6133.39	6800.72	3600.00	10
3	50	16546.10	16885.90	3600.00	2	6259.79	6881.77	3600.00	9
4	50	18247.60	18978.50	3600.00	4	6094.62	6850.73	3600.00	11
5	50	17356.30	17887.00	3600.00	3	5897.69	6921.84	3600.00	15

B.1.2 6-period instances

Table B.7: IRP results for 6 periods, 1 vehicle

		High inventory costs				Low inventory costs			
abs	$\|\mathcal{N}\|$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	t(s)	$gap(\%)$
1	5	5788.85	5789.35	1.22	0	3187.30	3187.30	0.63	0
2	5	4883.57	4883.77	0.77	0	2565.72	2565.92	1.27	0
3	5	6642.81	6643.29	2.92	0	4489.67	4489.83	2.49	0
4	5	5076.43	5076.88	1.34	0	3174.06	3174.35	2.04	0
5	5	4377.31	4377.71	1.21	0	2266.97	2267.10	0.90	0
1	10	8479.35	8480.17	18.52	0	4141.12	4141.53	22.83	0
2	10	8346.61	8347.44	96.19	0	5044.13	5044.63	76.86	0
3	10	8320.92	8321.68	9.42	0	4506.42	4506.83	11.23	0
4	10	8473.42	8474.26	10.99	0	4823.07	4823.53	11.73	0
5	10	9385.13	9386.03	9.85	0	4545.54	4545.98	9.73	0
1	15	11822.40	11823.50	440.51	0	5388.54	5389.08	742.15	0
2	15	13304.40	13305.70	21.42	0	5897.10	5897.68	35.43	0
3	15	12051.40	12052.60	1314.46	0	5417.93	5418.47	467.30	0
4	15	10478.20	10479.30	420.81	0	5334.48	5335.01	494.18	0
5	15	10053.10	10054.10	191.69	0	5052.01	5052.51	364.73	0
1	20	14265.10	14266.50	1426.21	0	5956.71	5957.31	1381.63	0
2	20	14317.90	14319.40	466.35	0	6783.38	6784.06	1019.41	0
3	20	14458.80	14477.80	3600.00	0	6113.43	6114.04	3037.08	0
4	20	16192.30	16194.00	536.52	0	7461.33	7462.08	1142.81	0
5	20	14042.60	14390.30	3600.00	2	7109.01	7310.80	3600.00	3
1	25	15292.30	15725.30	3600.00	3	6740.65	6968.58	3600.00	3
2	25	15166.40	15565.40	3600.00	3	6812.29	7090.19	3600.00	4
3	25	16287.90	16711.10	3600.00	3	7026.58	7257.15	3600.00	3
4	25	17639.10	17833.40	3600.00	1	7296.87	7614.55	3600.00	4
5	25	18372.90	18638.90	3600.00	1	6849.30	7120.70	3600.00	4
1	30	22546.90	23079.30	3600.00	2	7805.38	8195.27	3600.00	5
2	30	19560.00	19888.00	3600.00	2	7319.69	7631.21	3600.00	4
3	30	22991.20	23110.30	3600.00	1	7967.92	8203.69	3600.00	3
4	30	17424.80	17509.80	3600.00	0	7351.58	7508.28	3600.00	2
5	30	18460.30	18775.30	3600.00	2	7008.65	7231.97	3600.00	3

Table B.8: IRP results for 6 periods, 2 vehicles

		High inventory costs				Low inventory costs			
abs	$\lvert\mathcal{N}\rvert$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	t(s)	$gap(\%)$
1	5	6378.92	6379.56	2.63	0	3775.32	3775.68	4.15	0
2	5	5498.13	5498.64	3.19	0	3184.88	3185.19	5.09	0
3	5	8107.22	8108.03	30.69	0	3805.60	3805.98	16.80	0
4	5	5717.45	5718.02	11.50	0	2891.61	2891.85	3.45	0
5	5	4998.79	4999.24	1.14	0	5962.16	5962.76	142.26	0
1	10	9070.14	9071.05	201.20	0	5259.91	5260.44	210.61	0
2	10	10017.40	10018.40	42.03	0	5137.01	5137.52	45.03	0
3	10	9880.61	9996.96	3600.00	1	5528.78	5667.41	3600.00	2
4	10	9630.80	9814.04	3600.00	2	6323.21	6512.43	3600.00	3
5	10	9740.27	9864.35	3600.00	1	6034.72	6198.49	3600.00	3
1	15	12478.70	12637.70	3600.00	1	5816.69	6002.38	3600.00	3
2	15	12531.30	12568.50	3600.00	0	6085.22	6152.90	3600.00	1
3	15	14192.90	14450.10	3600.00	2	6726.91	7011.20	3600.00	4
4	15	11125.70	11267.90	3600.00	1	5923.92	6126.74	3600.00	3
5	15	11084.20	11375.40	3600.00	3	6066.40	6365.00	3600.00	5
1	20	15283.70	15622.10	3600.00	2	7111.92	7493.37	3600.00	5
2	20	14831.80	14955.00	3600.00	1	6281.58	6358.66	3600.00	1
3	20	14821.40	15025.30	3600.00	1	7277.66	7477.95	3600.00	3
4	20	14993.10	15450.80	3600.00	3	7920.91	8314.01	3600.00	5
5	20	16769.70	17162.30	3600.00	2	8182.95	8588.11	3600.00	5
1	25	15865.70	15955.70	3600.00	1	7419.45	7464.12	3600.00	1
2	25	17267.40	17591.50	3600.00	2	8030.15	8336.09	3600.00	4
3	25	18816.30	19034.60	3600.00	1	8492.33	8710.77	3600.00	3
4	25	16552.00	16663.10	3600.00	1	7761.35	7884.33	3600.00	2
5	25	19765.00	20003.70	3600.00	1	8249.37	8538.53	3600.00	3
1	30	23364.70	23911.40	3600.00	2	8585.37	9165.95	3600.00	6
2	30	20212.50	20683.80	3600.00	2	7951.33	8605.09	3600.00	8
3	30	23377.00	23416.80	3600.00	0	8387.87	8442.43	3600.00	1
4	30	17792.30	18317.40	3600.00	3	7736.92	8490.49	3600.00	9
5	30	18955.50	19645.90	3600.00	4	7465.74	8220.79	3600.00	9

Table B.9: IRP results for 6 periods, 3 vehicles

		High inventory costs				Low inventory costs			
abs	$\|\mathcal{N}\|$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	t(s)	$gap(\%)$
1	5	7258.32	7259.05	12.03	0	4656.69	4657.15	9.39	0
2	5	9861.91	9862.89	15.99	0	7702.77	7703.53	16.96	0
3	5	6405.10	6405.74	16.54	0	4494.86	4495.31	20.75	0
4	5	6529.70	6530.35	458.31	0	3878.70	3879.09	44.73	0
5	5	5972.18	5972.78	90.37	0	4220.58	4221.00	1113.03	0
1	10	9980.56	9981.56	776.02	0	5791.92	5792.50	417.49	0
2	10	10672.40	10673.50	228.28	0	6883.71	7103.26	3600.00	3
3	10	11223.40	11469.60	3600.00	2	7890.42	8097.79	3600.00	3
4	10	11205.30	11413.00	3600.00	2	6159.45	6185.17	3600.00	0
5	10	11008.10	11106.60	3600.00	1	7308.82	7417.08	3600.00	1
1	15	13325.10	13515.80	3600.00	1	6682.92	6861.16	3600.00	3
2	15	13364.10	13510.60	3600.00	1	6936.21	7113.75	3600.00	2
3	15	15248.20	15600.00	3600.00	2	7819.23	8160.36	3600.00	4
4	15	12043.90	12296.00	3600.00	2	6771.58	7267.16	3600.00	7
5	15	12338.90	12610.80	3600.00	2	7344.74	7606.32	3600.00	3
1	20	16620.40	16931.70	3600.00	2	8433.32	8782.16	3600.00	4
2	20	15371.00	15484.60	3600.00	1	6796.05	6928.07	3600.00	2
3	20	15728.50	15931.90	3600.00	1	8201.40	8387.05	3600.00	2
4	20	16491.40	17243.20	3600.00	4	9445.16	10116.80	3600.00	7
5	20	18591.00	19005.70	3600.00	2	10014.90	10498.80	3600.00	5
1	25	16544.90	16915.80	3600.00	2	8098.22	8417.94	3600.00	4
2	25	18498.40	19020.80	3600.00	3	9241.86	9728.38	3600.00	5
3	25	20213.20	20758.90	3600.00	3	9883.21	10359.40	3600.00	5
4	25	17230.80	17599.40	3600.00	2	8476.48	9107.76	3600.00	7
5	25	21323.20	21798.20	3600.00	2	9843.21	10464.20	3600.00	6
1	30	24484.60	25341.50	3600.00	3	9703.94	11003.80	3600.00	12
2	30	21225.20	22025.00	3600.00	4	8949.45	9626.14	3600.00	7
3	30	24009.90	24399.30	3600.00	2	9023.93	9453.25	3600.00	5
4	30	18723.90	19338.60	3600.00	3	8667.34	9713.45	3600.00	11
5	30	19947.30	20811.80	3600.00	4	8475.49	9779.33	3600.00	13

Table B.10: IRP results for 6 periods, 4 vehicles

		High inventory costs				Low inventory costs			
abs	$\|\mathcal{N}\|$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	t(s)	$gap(\%)$
1	5	8110.51	8111.31	13.91	0	5505.65	5506.19	17.47	0
2	5	7253.29	7254.01	14.34	0	4951.01	4951.50	17.65	0
3	5	6984.76	6985.40	2.13	0	9319.80	9320.73	30.26	0
4	5	11474.10	11475.30	88.76	0	5085.33	5085.81	2.89	0
5	5	6951.20	6951.89	27.05	0	4873.30	4873.79	187.63	0
1	10	12575.70	12748.80	3600.00	1	8245.87	8421.88	3600.00	2
2	10	13052.40	13181.90	3600.00	1	9748.58	9874.21	3600.00	1
3	10	11066.80	11067.90	3497.16	0	7215.35	7255.57	3600.00	1
4	10	12232.70	12323.90	3600.00	1	8513.30	8645.11	3600.00	2
5	10	11369.40	11471.50	3600.00	1	6489.47	6604.90	3600.00	2
1	15	14287.30	14410.80	3600.00	1	7624.39	7773.55	3600.00	2
2	15	14356.80	14515.50	3600.00	1	7926.37	8073.85	3600.00	2
3	15	16367.60	16709.10	3600.00	2	8914.69	9284.44	3600.00	4
4	15	13130.30	13465.00	3600.00	2	7979.88	8316.65	3600.00	4
5	15	13635.70	14011.00	3600.00	3	8633.65	9008.01	3600.00	4
1	20	18154.10	18526.10	3600.00	2	9939.81	10321.10	3600.00	4
2	20	15932.30	16157.20	3600.00	1	7380.92	7609.65	3600.00	3
3	20	16624.40	16988.60	3600.00	2	9128.29	9502.85	3600.00	4
4	20	18154.40	18762.30	3600.00	3	11051.70	11647.80	3600.00	5
5	20	20482.90	21078.50	3600.00	3	11872.20	12554.90	3600.00	5
1	25	17424.70	17841.80	3600.00	2	8930.54	9430.07	3600.00	5
2	25	19801.70	20653.80	3600.00	4	10542.00	11255.80	3600.00	6
3	25	21870.30	22635.40	3600.00	3	11527.70	12492.70	3600.00	8
4	25	18133.10	18649.70	3600.00	3	9331.65	9889.61	3600.00	6
5	25	23096.60	23870.00	3600.00	3	11573.30	12499.80	3600.00	7
1	30	25892.60	27150.40	3600.00	5	11132.80	12592.40	3600.00	12
2	30	22364.20	23091.70	3600.00	3	10080.60	10861.30	3600.00	7
3	30	24869.10	25465.60	3600.00	2	9855.42	10239.60	3600.00	4
4	30	19859.40	20733.80	3600.00	4	9832.02	10699.80	3600.00	8
5	30	21039.80	22026.30	3600.00	4	9561.87	11013.70	3600.00	13

Table B.11: IRP results for 6 periods, 5 vehicles

		High inventory costs				Low inventory costs			
abs	$\|\mathcal{N}\|$	z_{LB}	z_{UB}	$t(s)$	$gap(\%)$	z_{LB}	z_{UB}	t(s)	$gap(\%)$
1	5	9000.31	9001.21	24.56	0	6395.14	6395.78	21.81	0
2	5	8298.03	8298.86	225.81	0	11281.60	11282.70	4.88	0
3	5	13397.80	13399.10	5.40	0	6284.09	6284.72	134.94	0
4	5	8176.81	8177.63	119.52	0	6008.80	6009.40	1429.72	0
5	5	infeasible [1]							
1	10	14011.90	14143.50	3600.00	1	8185.35	8186.17	2153.92	0
2	10	14566.80	14933.10	3600.00	2	9672.66	9807.36	3600.00	1
3	10	11980.50	12005.80	3600.00	0	11261.20	11631.30	3600.00	3
4	10	13555.90	13752.00	3600.00	1	9889.12	10081.20	3600.00	2
5	10	12031.00	12068.50	3600.00	0	7178.28	7214.49	3600.00	1
1	15	15154.00	15446.10	3600.00	2	8503.58	8801.68	3600.00	3
2	15	15343.70	15476.90	3600.00	1	8919.41	9073.11	3600.00	2
3	15	17562.40	17842.70	3600.00	2	10070.50	10432.10	3600.00	3
4	15	14426.20	14540.50	3600.00	1	9268.88	9385.89	3600.00	1
5	15	14987.30	15437.70	3600.00	3	9937.35	10547.40	3600.00	6
1	20	19633.40	19995.50	3600.00	2	11439.00	11825.80	3600.00	3
2	20	16643.20	16870.50	3600.00	1	8064.15	8299.47	3600.00	3
3	20	17660.10	18073.10	3600.00	2	10153.70	10494.40	3600.00	3
4	20	19861.00	20359.60	3600.00	2	12707.00	13445.30	3600.00	5
5	20	22419.40	23116.00	3600.00	3	13904.90	14469.50	3600.00	4
1	25	18368.80	18874.90	3600.00	3	9860.04	10350.10	3600.00	5
2	25	21260.20	22038.40	3600.00	4	11959.00	12709.10	3600.00	6
3	25	23569.20	24422.10	3600.00	3	13224.00	14114.50	3600.00	6
4	25	19052.90	19571.80	3600.00	3	10264.20	10773.20	3600.00	5
5	25	24941.30	25711.40	3600.00	3	13454.90	14332.30	3600.00	6
1	30	23513.80	24369.30	3600.00	4	12633.40	13975.90	3600.00	10
2	30	25828.90	26273.90	3600.00	2	11273.50	12026.10	3600.00	6
3	30	21112.90	21824.60	3600.00	3	10819.20	11387.40	3600.00	5
4	30	22205.40	23224.90	3600.00	4	11067.70	11971.00	3600.00	8
5	30	27428.60	28511.80	3600.00	4	10751.30	12064.90	3600.00	11

[1] Infeasible because when dividing the total transportation capacity by the number of vehicles, each vehicle capacity is under the values expected to meet some customers demands.

Appendix C

HIRP-BS extended results

Go to the Table of Contents ↺
Go to the HIRP-BS results section ↺

In this Appendix, the results for the new set of instances to treat the HIRP-BS are presented. These instances were introduced in Subsection 2.4.2 of Chapter 2. To solve them, two approaches are considered: the linear flow formulation from Section 2.3, Chapter 2 and the SEMPO metaheuristic from Chapter 4.

C.1 Tables description

The instances are divided into three groups: Small, Medium and Large-scale instances. A summary for each table is given below.

Small-scale instances (S)

- **Table C.1.** Provides valid upper bounds as well as the linear relaxation values considering the mathematical formulation of the problem and the gap obtained when comparing the upper bounds and the metaheuristic solutions over ten runs.
- **Table C.2.** Gives valid upper bounds and the linear relaxation values considering the mathematical formulation of the problem and the gap obtained when comparing the linear relaxation values and the metaheuristic solutions over ten runs.
- **Table C.3.** Provides valid upper bounds as well as the linear relaxation values considering the mathematical formulation of the problem and time needed by the metaheuristic, in seconds, to obtain the best feasible solution over ten runs.
- **Table C.4.** Presents valid upper bounds as well as the linear relaxation values considering the mathematical formulation of the problem and presents some metrics including average, starndard deviation and gap when comparing the upper bounds and the metaheuristic firstly, and then the linear relaxation and the metaheuristic secondly.

Medium-scale instances (M)

- **Table C.5.** Presents the linear relaxation values from the mathematical formulation and compares the gap obtained when considering the metaheuristic valid best upper bounds over the ten runs.

- **Table C.6.** Presents the time to target, in seconds, needed by the metaheuristic to obtain the best valid upper bounds over each of the ten runs.
- **Table C.7.** Presents some metrics including the average values, the standard deviation as well as the representation in terms of percentage of the standard deviation when regarding the average solution value obtained by the metaheuristic.

Large-scale instances (L)

- **Table C.8.** Presents the linear relaxation values from the mathematical formulation and compares the gap obtained when considering the metaheuristic valid best upper bounds over the ten runs. note that for some instances, even the linear relaxation is not capable of finding a solution over one our of processing. Due to this, no comparison can be done with the metaheuristic feasible solutions.
- **Table C.9.** Presents the time to target, in seconds, needed by the metaheuristic to obtain the best valid upper bounds over each of the ten runs. Blank values also correspond to the instances for which a value is not known using the mathematical formulation.
- **Table C.10.** Presents some metrics including the average values, the standard deviation as well as the representation in terms of percentage of the standard deviation when regarding the average solution value obtained by the metaheuristic. Blank values also corresponds to the instances for which a value is not known using the mathematical formulation.

C.1.1 Small-scale instances

Table C.1: (S) UB and metaheuristic comparison

Instances			CPLEX							$Gap(\%)$ - z_{UB} and z_{Meta}									
$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	ID	z_{RL}	$t_{RL}(s)$	z_{LB}	z_{UB}	$gap(\%)$	$t_{target}(s)$	$t_{total}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
		1	14362.21	0.59	-	-	-	-	3600	-	-	-	-	-	-	-	-	-	-
		2	14349.82	0.75	-	-	-	-	3600	-	-	-	-	-	-	-	-	-	-
		3	13192.77	0.61	13468.21	13787.07	2.31	2755	3600	2.53	1.64	1.95	2.06	1.52	2.07	2.04	2.83	2.08	2.49
		4	15117.76	0.80	15410.03	16070.71	4.11	3600	3600	-0.79	-1.28	-1.30	-1.32	-0.92	-0.87	-1.09	-1.07	-1.26	-0.88
		5	12649.01	0.69	13048.71	14871.80	12.26	3600	3600	-0.85	-2.00	-2.78	-1.14	-2.70	-1.83	-2.25	-2.66	-1.85	-2.32
		6	12013.27	0.68	12365.08	12709.81	2.71	3600	3600	3.52	3.38	3.15	3.21	3.15	3.20	3.63	2.81	3.53	3.16
19	7	7	11091.07	0.47	11395.59	11667.71	2.33	3600	3600	2.07	2.35	1.93	1.95	1.59	1.99	2.13	2.10	2.19	2.07
		8	13417.14	0.85	13674.95	13948.05	1.96	3600	3600	4.40	3.79	4.75	4.65	4.72	4.13	4.88	3.85	4.53	3.85
		9	11195.59	0.55	11491.10	11722.88	1.98	3560	3600	2.29	2.09	2.08	2.10	1.93	2.22	2.11	2.17	2.25	2.13
		10	12406.95	0.73	12752.75	13327.03	4.31	3600	3600	0.47	0.12	0.32	-0.02	0.49	0.69	0.33	0.47	-0.01	0.65
		11	15287.84	0.80	15568.40	17630.62	11.70	3600	3600	-7.91	-7.74	-8.17	-7.82	-8.01	-8.20	-7.94	-7.95	-8.41	-8.35
		12	15118.76	0.74	15388.55	15689.12	1.92	3562	3600	2.77	2.62	2.44	2.45	2.60	2.81	2.19	2.57	2.72	2.65
		13	16058.63	0.77	16285.27	16535.68	1.51	3600	3600	2.10	2.76	2.85	2.71	2.75	2.46	2.60	3.00	2.28	2.93
							4.28	3519		0.96	0.70	0.66	0.80	0.65	0.79	0.79	0.74	0.73	0.76

Table C.2: (S) RL and metaheuristic comparison

Instances			CPLEX							$Gap(\%)$ - z_{RL} and z_{Meta}									
$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{T}\rvert$	ID	z_{RL}	$t_{RL}(s)$	z_{LB}	z_{UB}	$gap(\%)$	$t_{target}(s)$	$t_{total}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
		1	14362.21	0.59	-	-	-	-	3600	4.99	5.43	5.38	4.70	5.41	4.88	5.05	5.08	5.17	4.91
		2	14349.82	0.75	-	-	-	-	3600	7.33	7.55	7.22	7.15	7.74	7.05	7.00	7.26	7.13	7.39
		3	13192.77	0.61	13468.21	13787.07	2.31	2755	3600	6.73	5.88	6.18	6.28	5.77	6.29	6.26	7.02	6.31	6.69
		4	15117.76	0.80	15410.03	16070.71	4.11	3600	3600	5.19	4.72	4.71	4.68	5.07	5.11	4.90	4.92	4.74	5.10
		5	12649.01	0.69	13048.71	14871.80	12.26	3600	3600	14.23	13.24	12.58	13.97	12.65	13.39	13.03	12.68	13.37	12.98
		6	12013.27	0.68	12365.08	12709.81	2.71	3600	3600	8.81	8.67	8.46	8.52	8.46	8.51	8.91	8.14	8.82	8.47
19	7	7	11091.07	0.47	11395.59	11667.71	2.33	3600	3600	6.91	7.17	6.77	6.80	6.45	6.83	6.97	6.94	7.03	6.91
		8	13417.14	0.85	13674.95	13948.05	1.96	3600	3600	8.04	7.45	8.37	8.28	8.35	7.78	8.50	7.51	8.17	7.51
		9	11195.59	0.55	11491.10	11722.88	1.98	3560	3600	6.69	6.50	6.49	6.50	6.34	6.62	6.51	6.57	6.65	6.54
		10	12406.95	0.73	12752.75	13327.03	4.31	3600	3600	7.34	7.01	7.20	6.89	7.36	7.55	7.22	7.34	6.89	7.51
		11	15287.84	0.80	15568.40	17630.62	11.70	3600	3600	6.43	6.57	6.20	6.51	6.34	6.18	6.40	6.40	6.00	6.05
		12	15118.76	0.74	15388.55	15689.12	1.92	3562	3600	6.31	6.16	5.99	6.00	6.14	6.34	5.75	6.11	6.26	6.19
		13	16058.63	0.77	16285.27	16535.68	1.51	3600	3600	4.92	5.57	5.65	5.52	5.56	5.27	5.41	5.79	5.10	5.73
							4.28	3519		7.22	7.07	7.02	7.06	7.05	7.06	7.07	7.06	7.05	7.07

Table C.3: (S) UB and metaheuristic time to target comparison

Instances			CPLEX							$t_{target}(s)$ Metaheuristic									
$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	ID	z_{RL}	$t_{RL}(s)$	z_{LB}	z_{UB}	$gap(\%)$	$t_{target}(s)$	$t_{total}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
		1	14362.21	0.59	-	-	-	-	3600	764	803	1269	10	1303	1230	623	1079	799	10
		2	14349.82	0.75	-	-	-	-	3600	5	251	3	1522	501	1009	1191	10	519	1253
		3	13192.77	0.61	13468.21	13787.07	2.31	2755	3600	307	920	1090	928	1778	251	1631	2	929	1631
		4	15117.76	0.80	15410.03	16070.71	4.11	3600	3600	763	1517	1042	528	4	765	264	1491	750	1285
		5	12649.01	0.69	13048.71	14871.80	12.26	3600	3600	931	1434	1120	332	180	858	913	874	158	416
		6	12013.27	0.68	12365.08	12709.81	2.71	3600	3600	278	156	461	6	1050	1639	1582	1147	1600	1595
19	7	7	11091.07	0.47	11395.59	11667.71	2.33	3600	3600	1680	252	1462	673	281	1462	1304	230	1563	1724
		8	13417.14	0.85	13674.95	13948.05	1.96	3600	3600	254	1074	259	2	568	530	903	189	4	1675
		9	11195.59	0.55	11491.10	11722.88	1.98	3560	3600	761	1635	1483	1014	762	1589	8	1025	520	1145
		10	12406.95	0.73	12752.75	13327.03	4.31	3600	3600	1717	1319	1741	475	4	1136	896	460	486	1465
		11	15287.84	0.80	15568.40	17630.62	11.70	3600	3600	1610	1784	1094	1461	917	1798	1146	1443	1193	2
		12	15118.76	0.74	15388.55	15689.12	1.92	3562	3600	1112	508	260	1528	469	4	1464	189	975	652
		13	16058.63	0.77	16285.27	16535.68	1.51	3600	3600	520	401	185	697	1491	1591	5	7	714	744
							4.28	3519		823	927	882	706	716	1066	918	627	785	1046

Table C.4: (S) UB. RL and metaheuristic overview metrics

Instances			CPLEX							UB/Meta									RL/Meta		
$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	ID	z_{RL}	$t_{RL}(s)$	z_{LB}	z_{UB}	$gap(\%)$	$t_{target}(s)$	$t_{total}(s)$	avg	sdv	z_{UB}^{best}	$gap_{z_{UB}^{best}}$	z_{UB}^{worst}	$gap_{z_{UB}^{worst}}$	avg_{UB}	sdv_{UB}	Time(s)	avg	sdv	Best
		1	14362.21	0.59	-	-	-	-	3600	-	-	15070.70	-	15186.60	-	15134.38	39.25	789	5.10	0.25	4.70
		2	14349.82	0.75	-	-	-	-	3600	-	-	15430.10	-	15554.10	-	15477.01	38.34	626	7.28	0.23	7.00
		3	13192.77	0.61	13468.21	13787.07	2.31	2755	3600	2.12	0.40	14000.50	1.52	14188.70	2.83	14086.11	57.48	947	6.34	0.38	5.77
		4	15117.76	0.80	15410.03	16070.71	4.11	3600	3600	-1.08	0.20	15860.80	-1.32	15945.00	-0.79	15899.31	32.12	841	4.92	0.19	4.68
		5	12649.01	0.69	13048.71	14871.80	12.26	3600	3600	-2.04	0.65	14470.00	-2.78	14746.90	-0.85	14575.43	93.03	722	13.21	0.55	12.58
		6	12013.27	0.68	12365.08	12709.81	2.71	3600	3600	3.27	0.24	13077.60	2.81	13188.20	3.63	13140.11	32.91	951	8.58	0.23	8.14
19	7	7	11091.07	0.47	11395.59	11667.71	2.33	3600	3600	2.04	0.20	11855.70	1.59	11948.10	2.35	11910.31	24.30	1063	6.88	0.19	6.45
		8	13417.14	0.85	13674.95	13948.05	1.96	3600	3600	4.36	0.42	14497.40	3.79	14663.30	4.88	14583.48	63.23	546	8.00	0.40	7.45
		9	11195.59	0.55	11491.10	11722.88	1.98	3560	3600	2.14	0.10	11953.50	1.93	11997.70	2.29	11979.08	12.65	994	6.54	0.10	6.34
		10	12406.95	0.73	12752.75	13327.03	4.31	3600	3600	0.35	0.25	13324.60	-0.02	13420.20	0.69	13374.03	34.05	970	7.23	0.24	6.89
		11	15287.84	0.80	15568.40	17630.62	11.70	3600	3600	-8.05	0.22	16263.50	-8.41	16363.60	-7.74	16317.16	33.57	1245	6.31	0.19	6.00
		12	15118.76	0.74	15388.55	15689.12	1.92	3562	3600	2.58	0.18	16041.20	2.19	16142.80	2.81	16105.04	30.15	716	6.12	0.18	5.75
		13	16058.63	0.77	16285.27	16535.68	1.51	3600	3600	2.64	0.29	16889.80	2.10	17046.40	3.00	16984.97	50.20	636	5.45	0.28	4.92
							4.28	3519		0.76	0.29		0.31		1.19		41.64	850	7.07	0.26	6.67

C.1.2 Medium-scale instances

Table C.5: (M) RL and metaheuristic comparison

Instances			CPLEX		*Gap*(%) - z_{RL} and z_{Meta}									
$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{T}\rvert$	ID	z_{RL}	$t_{RL}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
19	14	1	31765.05	1.37	6.98	7.37	6.19	7.42	6.67	7.26	6.91	6.45	6.84	7.34
		2	35995.55	1.14	5.68	5.43	5.37	5.73	5.41	5.42	5.64	5.52	5.52	5.47
		3	36849.87	1.57	6.69	6.96	6.59	6.99	6.83	6.82	6.78	6.45	6.62	7.13
	21	1	70705.51	1.77	4.96	4.95	4.95	5.09	5.28	5.10	4.86	5.33	4.95	5.21
		2	67279.76	1.58	6.08	6.23	6.24	6.50	6.30	6.36	6.17	6.28	6.22	6.22
		3	81136.08	1.61	5.40	5.45	5.68	5.77	5.68	5.88	5.26	5.84	5.83	5.29
	28	1	105572.10	2.73	6.19	7.46	7.47	6.50	7.42	5.95	6.07	7.52	6.27	7.31
		2	108853.94	2.52	5.74	6.77	5.71	5.92	5.80	5.46	5.97	5.86	5.48	5.38
		3	96870.65	2.50	5.99	7.02	6.19	5.96	5.85	5.75	6.23	6.99	6.19	6.20
			Avg. group gap(%)		5.97	6.40	6.04	6.21	6.14	6.00	5.99	6.25	5.99	6.17
34	7	1	21504.21	4.03	7.94	7.60	8.20	7.88	7.85	8.35	8.12	8.32	7.67	7.99
		2	25220.85	3.50	6.95	6.30	7.20	7.35	6.97	7.91	7.31	7.65	7.40	6.61
		3	19551.36	3.08	10.13	9.56	9.69	10.17	10.21	9.53	9.83	10.21	9.64	10.01
	14	1	64725.84	4.74	5.16	5.49	5.70	5.74	5.68	5.54	5.52	5.22	5.51	5.26
		2	58026.86	8.71	9.43	9.65	9.76	9.74	9.66	9.32	9.65	9.76	9.12	9.45
		3	60027.35	7.40	12.76	11.93	10.28	10.92	11.89	11.47	10.81	13.52	11.28	12.21
	21	1	112811.33	14.25	7.35	6.31	6.93	6.53	7.06	6.66	6.43	6.71	7.17	6.51
		2	142540.13	12.29	7.59	7.59	7.11	7.57	6.37	7.64	7.64	6.46	7.51	7.01
		3	125961.64	16.06	7.99	7.05	7.89	8.07	7.99	7.69	7.71	7.25	7.71	7.62
	28	1	199983.60	14.58	5.46	5.25	5.37	5.02	5.34	5.20	5.44	5.40	5.53	5.60
		2	194899.21	18.19	8.09	8.20	8.18	8.15	8.12	8.04	8.12	8.51	8.12	8.25

Continued on next page

Table C.5 – (M) RL and metaheuristic comparison (continued)

Instances			CPLEX		*Gap*(%) - z_{RL} and z_{Meta}									
$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	ID	z_{RL}	$t_{RL}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
		3	198494.93	18.27	6.60	6.53	7.25	6.79	7.24	7.14	6.77	6.38	6.58	7.09
		Avg. group gap(%)			7.96	7.62	7.80	7.83	7.86	7.87	7.78	7.95	7.77	7.80
46	7	1	28952.92	9.11	8.80	10.13	9.50	8.18	8.65	8.90	9.39	8.90	8.91	9.19
		2	31393.56	7.63	9.06	9.61	9.81	9.78	9.06	9.83	10.31	9.95	9.98	8.90
		3	25408.63	4.75	8.09	8.31	8.35	8.20	8.07	8.04	8.75	8.47	7.83	8.07
	14	1	70858.71	14.24	5.90	6.20	5.76	5.84	6.07	5.65	5.27	5.88	5.66	5.89
		2	85473.87	19.85	6.72	6.78	6.57	6.83	6.84	6.77	6.86	6.79	6.66	6.93
		3	78769.42	13.58	8.94	8.27	8.67	7.92	8.80	8.56	8.46	8.37	8.47	8.61
	21	1	163482.68	49.02	6.13	6.13	6.07	6.11	6.14	6.19	6.16	6.26	6.13	6.08
		2	165078.23	47.82	5.52	4.75	5.33	5.48	4.66	5.65	5.44	4.99	5.57	5.65
		3	129226.16	32.59	7.17	7.14	7.17	7.14	7.15	6.98	7.14	7.26	7.08	7.20
	28	1	232277.48	56.34	5.29	5.31	5.38	5.38	5.41	5.40	5.29	5.37	5.36	5.24
		2	263986.86	62.94	6.38	6.35	6.33	6.54	6.37	6.14	6.55	6.24	6.23	6.43
		3	249555.83	59.98	5.88	5.88	6.03	6.06	6.01	6.00	5.90	6.02	6.05	5.98
		Avg. group gap(%)			6.99	7.07	7.08	6.95	6.94	7.01	7.13	7.04	7.00	7.01
58	7	1	46315.26	24.86	12.31	13.00	12.53	12.14	12.38	12.70	12.14	13.15	13.25	12.94
		2	27474.70	21.40	20.06	19.00	19.44	18.64	20.06	19.53	19.51	18.99	19.45	19.75
		3	37506.26	18.19	15.14	15.21	14.99	15.08	15.07	15.27	14.90	14.01	14.19	14.27
	14	1	100261.99	32.61	11.52	11.23	11.35	11.34	11.30	11.02	11.18	10.96	11.18	10.99
		2	102989.69	29.79	17.19	17.24	17.27	17.54	17.08	17.04	16.95	17.04	17.03	17.33
		3	118288.86	20.56	10.47	8.95	10.40	9.54	10.31	9.96	10.40	10.09	8.92	10.15
	21	1	199673.24	78.55	10.30	10.40	10.26	10.41	10.35	10.75	10.83	10.32	10.40	10.35
		2	198632.41	65.69	10.70	10.61	10.34	10.16	10.33	10.61	10.05	10.46	10.24	10.50

Continued on next page

Table C.5 – (M) RL and metaheuristic comparison (continued)

Instances			CPLEX		$Gap(\%)$ - z_{RL} and z_{Meta}									
$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	ID	z_{RL}	$t_{RL}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
		3	187080.89	95.89	13.84	13.55	13.87	13.97	13.87	13.77	13.60	13.88	13.73	13.54

Continued on next page

Table C.5 – (M) RL and metaheuristic comparison (continued)

Instances			CPLEX		$Gap(\%)$ - z_{RL} and z_{Meta}									
$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	ID	z_{RL}	$t_{RL}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
	28	1	312777.77	86.24	7.63	8.34	8.24	8.25	8.30	8.13	7.98	7.85	7.96	8.02
		2	289651.41	134.67	10.80	11.10	10.96	11.00	10.94	10.77	10.67	10.78	10.76	10.73
		3	344415.46	120.88	9.95	9.73	10.03	9.66	9.97	9.67	9.59	10.68	10.09	10.23
			Avg. group gap(%)		12.49	12.36	12.47	12.31	12.50	12.44	12.32	12.35	12.27	12.40
83	7	1	46176.09	192.97	14.66	14.73	14.62	14.37	15.05	14.43	14.45	14.56	14.52	14.36
		2	45115.88	37.98	15.82	15.94	15.93	16.17	15.99	16.16	15.48	15.83	15.90	15.95
		3	48785.67	151.18	12.14	12.70	12.31	12.00	12.50	12.07	12.41	12.72	12.64	11.72
	14	1	150855.25	94.43	10.39	10.40	10.60	10.58	10.66	10.69	10.54	10.29	10.61	10.62
		2	115554.70	332.01	11.79	11.09	11.46	11.73	11.41	11.75	11.30	11.55	11.17	11.22
		3	161514.80	349.82	10.29	10.45	10.28	10.19	10.31	10.32	10.47	10.55	10.42	10.38
	21	1	220407.99	405.76	9.28	8.69	8.29	8.60	9.00	8.80	8.83	8.74	8.52	8.79
		2	272141.83	281.30	8.96	9.17	9.63	8.81	9.33	9.00	8.85	9.21	9.31	9.23
		3	254559.92	546.52	9.79	9.67	9.79	9.72	9.34	9.45	9.68	10.01	9.81	9.39
	28	1	-	-	-	-	-	-	-	-	-	-	-	-
		2	458910.42	618.15	8.10	8.21	8.57	8.38	8.23	8.30	8.20	8.42	8.49	8.34
		3	-	-	-	-	-	-	-	-	-	-	-	-
			Avg. group gap(%)		11.12	11.11	11.15	11.05	11.18	11.10	11.02	11.19	11.14	11.00
			Avg. global gap(%)		8.99	8.97	8.98	8.94	8.99	8.96	8.92	9.02	8.90	8.95

Table C.6: (M) UB and metaheuristic time to target comparison

Instances			CPLEX		t_{target} Metaheuristic									
$\mathcal{N}$	$\mathcal{T}$	ID	z_{RL}	$t_{RL}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
19	14	1	31765.05	1.37	553	806	1041	520	20	555	268	560	1080	275
		2	35995.55	1.14	1313	1042	1690	2	1273	1228	671	719	1152	255
		3	36849.87	1.57	710	1177	1300	1316	62	675	674	693	1052	789
	21	1	70705.51	1.77	33	18	1270	561	12	573	1776	1191	886	21
		2	67279.76	1.58	281	17	1127	1408	812	850	1111	283	1072	13
		3	81136.08	1.61	1409	620	1550	586	1061	1755	634	605	1163	1311
	28	1	105572.10	2.73	1371	1056	1619	1334	1405	905	47	1607	1612	1173
		2	108853.94	2.52	426	1179	230	1180	1117	325	303	1301	423	743
		3	96870.65	2.50	1052	177	1025	1442	69	1234	341	178	1007	957
			Avg. group time(s)		794	677	1206	928	648	900	647	793	1050	615
34	7	1	21504.21	4.03	603	879	656	1633	16	47	1732	646	1732	1339
		2	25220.85	3.50	48	47	1607	693	1056	1688	36	1052	625	1575
		3	19551.36	3.08	897	925	389	284	1262	326	991	350	632	289
	14	1	64725.84	4.74	120	1572	1124	1136	1457	1660	1799	1250	805	830
		2	58026.86	8.71	9	950	16	649	326	9	676	7	390	1675
		3	60027.35	7.40	107	845	1707	1030	969	773	406	1461	482	1671
	21	1	112811.33	14.25	892	135	809	840	1313	97	52	1181	1663	902
		2	142540.13	12.29	1469	1260	1422	66	112	1690	852	1688	941	691
		3	125961.64	16.06	1685	1411	1043	33	15	1659	1412	1363	13	773
	28	1	199983.60	14.58	64	98	1507	732	1558	1647	406	1285	389	1799
		2	194899.21	18.19	340	1749	168	884	427	1517	1723	1784	427	117
		3	198494.93	18.27	1139	1453	638	1347	1748	1179	1123	404	1592	1645
			Avg. group time(s)		614	943	924	777	855	1024	934	1039	807	1109

Continued on next page

Table C.6 – (M) UB and metaheuristic time to target comparison (continued)

Instances			CPLEX		t_{target} Metaheuristic									
$\mathcal{N}$	$\mathcal{T}$	ID	z_{RL}	$t_{RL}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
46	7	1	28952.92	9.11	1708	1091	155	1170	871	1012	1421	1655	1159	459
		2	31393.56	7.63	1411	1027	360	48	74	1458	1657	1073	48	568
		3	25408.63	4.75	1521	86	185	1522	93	898	472	1557	473	1526
	14	1	70858.71	14.24	1502	1718	678	225	606	331	1011	1447	1796	1272
		2	85473.87	19.85	128	357	1793	499	200	1739	207	1259	843	990
		3	78769.42	13.58	1018	161	608	71	923	721	1322	305	1604	1769
	21	1	163482.68	49.02	243	1669	1613	929	459	1485	796	373	142	1735
		2	165078.23	47.82	1685	423	1261	1176	39	177	1359	1618	1670	1472
		3	129226.16	32.59	1403	1323	1422	473	456	376	536	1498	316	1073
	28	1	232277.48	56.34	1212	535	1121	165	845	1765	1691	334	820	388
		2	263986.86	62.94	1798	1609	998	364	1203	511	591	879	431	980
		3	249555.83	59.98	799	926	1120	1625	523	767	891	1730	1046	1108
			Avg. group time(s)		1202	910	943	689	524	937	996	1144	862	1112
58	7	1	46315.26	24.86	390	429	1123	439	1652	1206	526	8	544	444
		2	27474.70	21.40	1115	1549	1121	433	756	1454	849	1468	1078	533
		3	37506.26	18.19	1194	1223	1208	488	82	854	583	102	1170	937
	14	1	100261.99	32.61	1505	702	318	685	1772	258	371	695	1419	1779
		2	102989.69	29.79	1063	351	76	1511	473	300	1091	950	715	751
		3	118288.86	20.56	471	1741	1724	1780	1123	1585	1087	1146	132	855
	21	1	199673.24	78.55	1599	235	1723	1783	1211	1157	1256	1714	1292	1107
		2	198632.41	65.69	896	169	528	1434	487	1401	1775	273	916	1221
		3	187080.89	95.89	98	1054	1143	195	991	180	1704	1194	168	1012

Continued on next page

Table C.6 – (M) UB and metaheuristic time to target comparison (continued)

Instances			CPLEX		t_{target} Metaheuristic									
$\mathcal{N}$	$\mathcal{T}$	ID	z_{RL}	$t_{RL}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
	28	1	312777.77	86.24	896	1137	1656	900	1584	447	1233	610	794	671
		2	289651.41	134.67	747	1597	1246	1003	705	991	1094	1238	1004	1291
		3	344415.46	120.88	384	850	510	1317	1754	636	278	352	523	1614
			Avg. group time(s)		863	920	1031	997	1049	872	987	813	813	1018
83	7	1	46176.09	192.97	211	1782	1163	204	1399	111	1158	966	97	921
		2	45115.88	37.98	838	208	1591	283	590	298	1034	886	125	931
		3	48785.67	151.18	1189	949	1629	1502	700	619	1319	706	413	942
	14	1	150855.25	94.43	1466	1720	83	244	992	575	680	1312	569	1099
		2	115554.70	332.01	757	1611	674	616	653	814	1115	383	641	858
		3	161514.80	349.82	410	535	311	1249	836	1305	1040	466	203	1758
	21	1	220407.99	405.76	1475	489	1725	1323	635	1670	1746	1064	1155	970
		2	272141.83	281.30	966	766	1105	1166	1754	1754	751	659	1699	327
		3	254559.92	546.52	945	1527	1476	1106	1501	1424	1671	1794	951	1329
	28	1	-	-	-	-	-	-	-	-	-	-	-	-
		2	458910.42	618.15	1318	128	562	1355	284	825	1122	966	278	1795
		3	-	-	-	-	-	-	-	-	-	-	-	-
			Avg. group time(s)		957	971	1032	904	934	939	1163	920	613	1093
			Avg. global time(s)		889	893	1017	854	806	936	954	951	825	1006

Table C.7: (M) RL and metaheuristic overview metrics

Instances $\mathcal{N}$	$\mathcal{T}$	ID	z_{best}	z_{worst}	z_{avg}	z_{std}	$z_{std}(\%)$	$t_{avg}(s)$
19	14	1	33861.00	34310.40	34135.72	152.40	0.45	568
		2	38037.40	38184.90	38098.06	50.75	0.13	934
		3	39391.00	39677.50	39533.20	87.06	0.22	845
	21	1	74315.90	74685.10	74477.94	125.89	0.17	634
		2	71637.70	71959.80	71773.83	86.61	0.12	697
		3	85644.50	86204.60	85957.79	213.05	0.25	1069
	28	1	112247.97	114155.00	113299.10	813.64	0.72	1213
		2	115040.73	116761.00	115569.96	486.08	0.42	723
		3	102777.00	104181.00	103317.85	482.34	0.47	748
34	7	1	23272.60	23462.20	23372.28	64.95	0.28	928
		2	26918.00	27388.40	27168.72	138.79	0.51	843
		3	21610.70	21773.50	21699.36	66.91	0.31	634
	14	1	68249.90	68664.50	68480.09	148.40	0.22	1175
		2	63853.30	64305.60	64156.87	152.21	0.24	471
		3	66903.30	69413.60	67992.80	749.34	1.10	945
	21	1	120413.00	121767.00	121000.50	449.45	0.37	788
		2	152232.00	154337.00	153684.00	812.55	0.53	1019
		3	135518.00	137021.00	136466.10	484.19	0.35	941
	28	1	210551.00	211847.00	211313.40	379.20	0.18	948
		2	211936.00	213026.00	212255.30	303.29	0.14	914
		3	212032.48	214014.00	213063.18	730.89	0.34	1227
46	7	1	31531.70	32216.50	31836.35	186.75	0.59	1070
		2	34461.50	35002.90	34739.92	179.66	0.52	772
		3	27565.80	27845.20	27684.01	79.09	0.29	833
	14	1	74801.30	75545.10	75232.20	203.08	0.27	1058
		2	91480.70	91840.00	91686.45	103.67	0.11	802
		3	85545.30	86500.70	86094.61	267.94	0.31	850
	21	1	174044.00	174392.00	174175.90	99.85	0.06	945
		2	173148.00	174971.00	174330.50	679.94	0.39	1088
		3	138930.00	139341.00	139166.00	108.66	0.08	888
	28	1	245113.00	245555.00	245387.30	148.83	0.06	888
		2	281252.00	282505.00	281903.30	396.04	0.14	936
		3	265153.00	265668.00	265435.40	191.95	0.07	1053

Continued on next page

Table C.7 - (M) RL and metaheuristic overview metrics (continued)

Instances $\mathcal{N}$	$\mathcal{T}$	ID	z_{best}	z_{worst}	z_{avg}	z_{std}	$z_{std}(\%)$	$t_{avg}(s)$
58	7	1	52713.20	53387.20	53026.60	251.08	0.47	676
		2	33768.10	34370.00	34107.19	194.95	0.57	1036
		3	43614.60	44267.90	44028.97	242.42	0.55	784
	14	1	112607.00	113317.00	112918.60	228.33	0.20	950
		2	124007.00	124892.00	124339.50	266.90	0.21	728
		3	129879.00	132119.00	131319.70	847.64	0.65	1164
	21	1	222510.00	223928.00	222944.20	479.55	0.22	1308
		2	220827.00	222426.00	221688.20	527.22	0.24	910
		3	216367.00	217453.00	216936.00	383.59	0.18	774
	28	1	338614.00	341230.00	340238.10	831.13	0.24	993
		2	324261.00	325817.00	324912.10	505.19	0.16	1091
		3	380963.00	385579.00	382520.00	1399.66	0.37	822
83	7	1	53916.50	54359.30	54054.96	132.68	0.25	801
		2	53376.90	53819.60	53657.34	124.79	0.23	679
		3	55261.00	55892.40	55641.00	211.91	0.38	997
	14	1	168151.00	168906.00	168624.30	251.30	0.15	874
		2	129975.00	131003.00	130494.40	370.69	0.28	812
		3	179839.00	180555.00	180194.70	214.70	0.12	811
	21	1	240328.00	242961.00	241556.80	713.51	0.30	1225
		2	298430.00	301140.00	299549.70	826.73	0.28	1095
		3	280787.00	282879.00	281799.80	659.61	0.23	1372
	28	1	485412.00	488434.00	487275.20	924.85	0.19	-
		2	499365.00	501946.00	500579.90	798.14	0.16	863
		3	442948.00	444221.00	443590.80	420.45	0.09	-

C.1.3 Large-scale instances

Table C.8: (L) RL and metaheuristic comparison

Instances			CPLEX		*Gap*(%) - z_{RL} and z_{meta}									
$\|\mathcal{N}\|$	$\|\mathcal{T}\|$	ID	z_{RL}	$t_{RL}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
114	7	1	63214.61	148.56	14.66	14.22	14.66	14.60	14.91	14.69	15.07	15.62	14.33	14.36
114	7	2	62803.88	92.41	13.83	14.13	14.03	13.13	13.56	13.51	13.97	14.00	14.24	14.11
114	7	3	62229.09	135.55	12.38	12.16	12.51	12.28	12.17	12.33	12.31	12.09	12.57	12.14
114	14	1	160846.19	313.28	12.64	12.14	12.39	12.39	12.64	12.14	12.64	12.29	12.66	12.89
114	14	2	240633.57	164.75	10.95	11.33	10.97	10.98	10.52	10.60	11.50	11.14	10.73	11.25
114	21	1	368429.18	1234.34	9.95	10.07	10.29	9.97	10.16	9.95	9.97	10.29	10.16	9.97
149	28	1	-	-	-	-	-	-	-	-	-	-	-	-
170	21	1	-	-	-	-	-	-	-	-	-	-	-	-
170	28	1	-	-	-	-	-	-	-	-	-	-	-	-
183	7	1	-	-	-	-	-	-	-	-	-	-	-	-
			Average gap(%)		12.40	12.34	12.48	12.23	12.33	12.20	12.58	12.57	12.45	12.45

Table C.9: (L) UB and metaheuristic time to target comparison

Instances			CPLEX		t_{target} SEMPO (s)									
$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{T}\rvert$	ID	z_{RL}	$t_{RL}(s)$	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
114	7	1	63214.61	148.56	1481	1418	914	1215	1689	1535	1237	1455	763	1169
114	7	2	62803.88	92.41	1106	849	798	939	1127	395	1599	1025	1308	1426
114	7	3	62229.09	135.55	1296	807	335	1355	920	1676	300	667	699	310
114	14	1	160846.19	313.28	618	312	956	756	710	1148	901	348	1127	11
114	14	2	240633.57	164.75	605	274	1241	474	1678	599	1222	867	1083	45
114	21	1	368429.18	1234.34	530	42	108	1699	126	555	1501	301	424	1545
149	28	1	-	-	-	-	-	-	-	-	-	-	-	-
170	21	1	-	-	-	-	-	-	-	-	-	-	-	-
170	28	1	-	-	-	-	-	-	-	-	-	-	-	-
183	7	1	-	-	-	-	-	-	-	-	-	-	-	-
			Avg. time(s)		940	618	726	1074	1042	985	1127	778	901	751

Table C.10: (L) LR and metaheuristic overview metrics

Instances			CPLEX		SEMPO				
$\mathcal{N}$	$\mathcal{T}$	ID	z_{RL}	$t_{RL}(s)$	z_{best}	z_{worst}	z_{avg}	z_{std}	$t_{avg}(s)$
114	7	1	63214.61	148.56	73697.50	74914.70	74120.87	357.73	1288.13
114	7	2	62803.88	92.41	72296.70	73236.40	72902.97	293.43	1057.53
114	7	3	62229.09	135.55	70788.60	71175.00	70952.09	129.24	836.96
114	14	1	160846.19	313.28	183068.00	184650.00	183790.90	524.65	689.14
114	14	2	240633.57	164.75	268914.00	271892.00	270369.50	965.68	809.31
114	21	1	368429.18	1234.34	409142.00	410697.00	409722.40	627.68	683.57
149	28	1	-	-	900932.00	905981.00	904061.50	2204.26	-
170	21	1	-	-	537235.00	539239.00	538574.90	687.44	-
170	28	1	-	-	852666.00	856599.00	854943.00	1628.57	-
183	7	1	-	-	98708.70	99711.30	99123.17	326.58	-
				Avg. time(s)				774.53	894.10

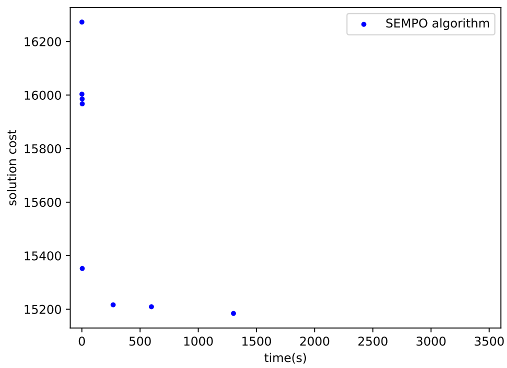
Appendix D

HIRP-BS convergence analysis

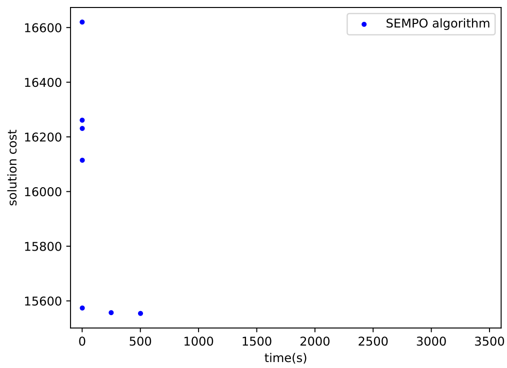
Go to the Table of Contents
Go to the HIRP-BS results section

The following graphics illustrate the convergence of the metaheuristic (blue dots) and the mathematical formulation (orange dots) for the Heterogeneous Inventory Routing Problem with Batch Size (HIRP-BS) over one hour (3600 seconds) of execution. The X axis contains the time and the Y the solution value found. The instances considered here are the thirteen small-scale ones (introduced in Chapter 2, Subsection 2.4.2).

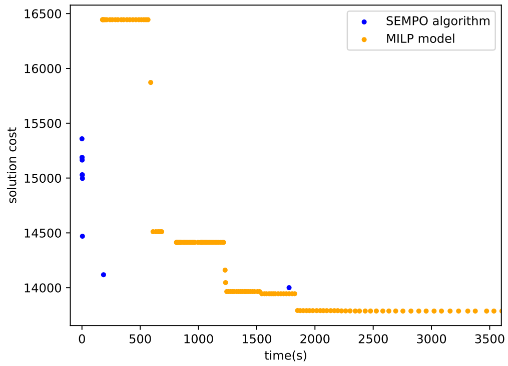
Instance `s_19_7_1`



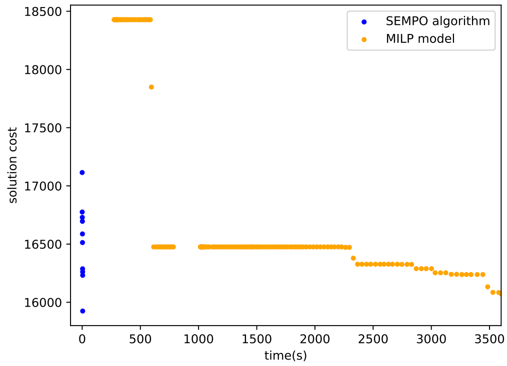
Instance `s_19_7_2`



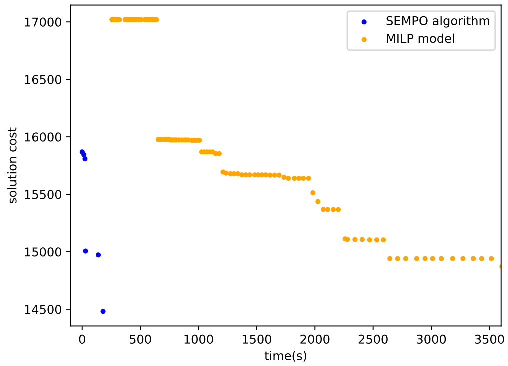
Instance s_19_7_3



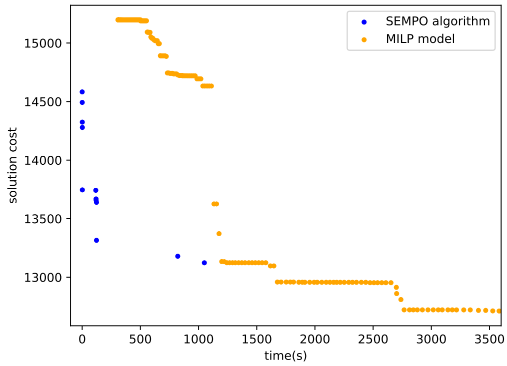
Instance s_19_7_4



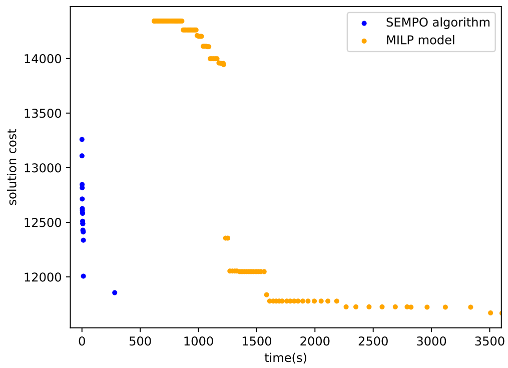
Instance s_19_7_5



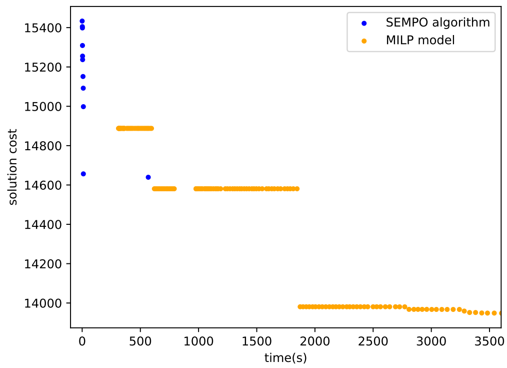
Instance s_19_7_6



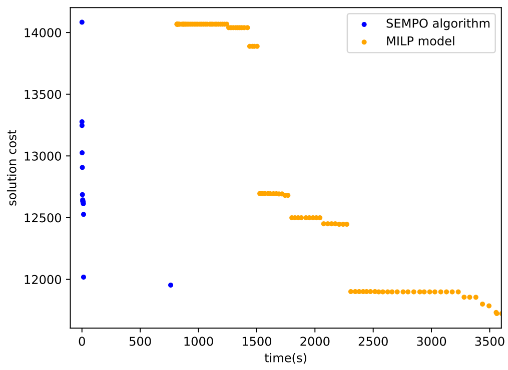
Instance s_19_7_7



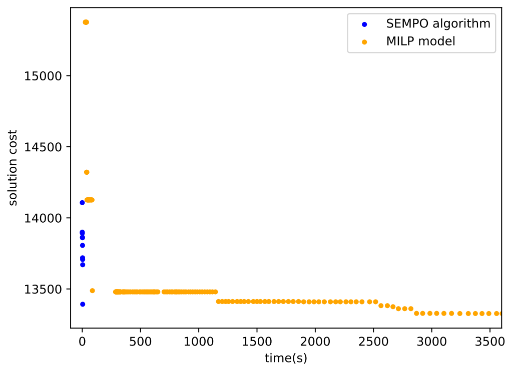
Instance s_19_7_8



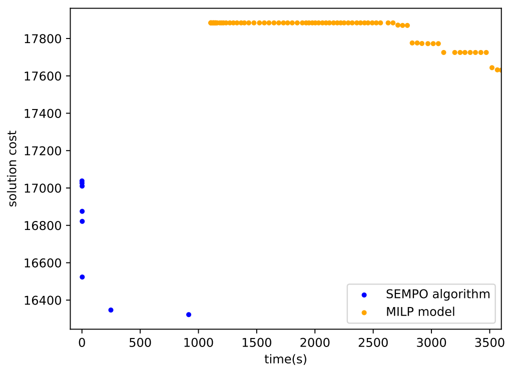
Instance s_19_7_9



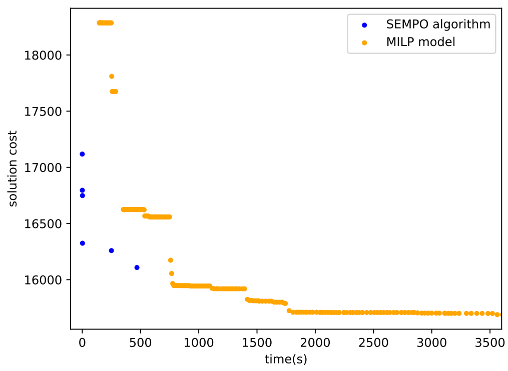
Instance s_19_7_10



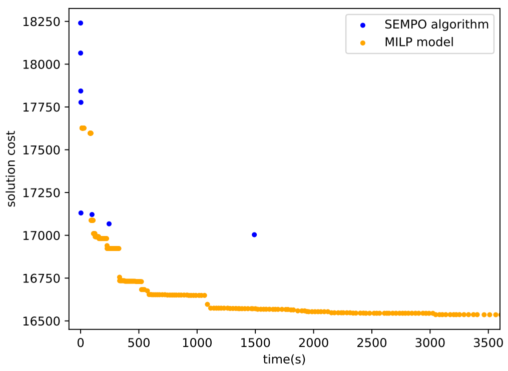
Instance s_19_7_11



Instance s_19_7_12



Instance s_19_7_13



Acronyms

ALNS Adaptive Large Neighborhood Search.

API Application Programming Interface.

BS Batch size inventory policy.

CARP Capacitated Arc Routing Problem.

CVRP Capacitated Vehicle Routing Problem.

ELS Evolutionary Local Search.

FSMVRP Fleet Size and Mix Vehicle Routing Problem.

GRASP Greedy Randomized Adaptive Search Procedure.

GVRP General Vehicle Routing Problem.

HIRP-BS Heterogeneous Inventory Routing Problem with Batch Size.

HVRP Heterogeneous Inventory Routing Problem.

IRP Inventory Routing Problem.

LB Lower Bound.

LR Linear Relaxation.

MILP Mixed Integer Linear Problem.

ML Maximum level inventory policy.

MMIRP Multi-product Multi-vehicle Inventory Routing Problem.

OR Operations Research.

OU Order-up-to level inventory policy.

PIRP Perishable Inventory Routing Problem.

RNVD Randomized Variable Neighbourhood Descent.

SEMPO Split-Embedded Metaheuristic with a Post-Optimization phase.

UB Upper Bound.

VRP Vehicle Routing Problem.

ZIO Zero-Inventory-Ordering inventory policy.

Bibliography

Aksen, D., Kaya, O., Salman, F. S., and Tüncel, Ö. (2014). An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. *European Journal of Operational Research*, 239(2):413–426. (Cited on pages 44, 51, and 55)

Alinaghian, M., Tirkolaee, E. B., Dezaki, Z. K., Hejazi, S. R., and Ding, W. (2021). An augmented tabu search algorithm for the green inventory-routing problem with time windows. *Swarm and evolutionary computation*, 60:100802. (Cited on page 45)

Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., and Morabito, R. (2020). Formulations, branch-and-cut and a hybrid heuristic algorithm for an inventory routing problem with perishable products. *European Journal of Operational Research*, 283(2):511–529. (Cited on pages 20, 45, 51, and 55)

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., and Løkketangen, A. (2010). Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536. (Cited on page 20)

Archetti, C., Bertazzi, L., Hertz, A., and Speranza, M. G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1):101–116. (Cited on pages 39, 51, 55, 69, 121, 143, and 148)

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. (Cited on pages 15, 21, 33, 34, 38, 44, 51, 55, 68, 69, 82, 121, 122, 129, 143, and 148)

Archetti, C., Bianchessi, N., Irnich, S., and Speranza, M. G. (2014). Formulations for an inventory routing problem. *International Transactions in Operational Research*, 21(3):353–374. (Cited on pages 21, 59, and 63)

Archetti, C., Christiansen, M., and Speranza, M. G. (2018). Inventory routing with pickups and deliveries. *European Journal of Operational Research*, 268(1):314–324. (Cited on pages 19, 51, and 55)

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. (Cited on pages 44, 51, and 55)

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. (Cited on pages 19, 20, 46, 51, and 55)

Baker, B. M. and Ayechew, M. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5):787–800. (Cited on page 38)

Beasley, J. E. (1983). Route first—cluster second methods for vehicle routing. *Omega*, 11(4):403–408. (Cited on page 47)

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. (Cited on pages 5, 7, 9, 19, 24, and 135)

Bertazzi, L., Coelho, L. C., Maio, A. D., and Laganà, D. (2019). A matheuristic algorithm for the multi-depot inventory routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 122:524–544. (Cited on pages 20, 51, and 55)

Boudia, M., Prins, C., and Reghioui, M. (2007). An effective memetic algorithm with population management for the split delivery vehicle routing problem. In *Hybrid Metaheuristics: 4th International Workshop, HM 2007, Dortmund, Germany, October 8-9, 2007. Proceedings 4*, pages 16–30. Springer. (Cited on page 107)

Bourreau, E., Gondran, M., Lacomme, P., and Vinot, M. (2019). *De la programmation linéaire à la programmation par contraintes*. Ellipses. (Cited on pages 79, 141)

Bourreau, E., Gondran, M., Lacomme, P., and Vinot, M. (2020). *Programmation Par Contraintes: démarches de modélisation pour des problèmes d'optimisation*. Ellipses. (Cited on pages 79, 141)

Charaf, S., Taş, D., Flapper, S. D. P., and Van Woensel, T. (2024a). A branch-and-price algorithm for the two-echelon inventory-routing problem. *Computers & Industrial Engineering*, page 110463. (Cited on pages 47, 52, and 55)

Charaf, S., Taş, D., Flapper, S. D. P., and Van Woensel, T. (2024b). A matheuristic for the two-echelon inventory-routing problem. *Computers & Operations Research*, 171:106778. (Cited on pages 47, 52, and 55)

Chassaing, M., Duhamel, C., and Lacomme, P. (2016). An els-based approach with dynamic probabilities management in local search for the dial-a-ride problem. *Engineering Applications of Artificial Intelligence*, 48:119–133. (Cited on page 117)

Cheng, C., Yang, P., Qi, M., and Rousseau, L.-M. (2017). Modeling a green inventory routing problem with a heterogeneous fleet. *Transportation Research Part E: Logistics and Transportation Review*, 97:97–112. (Cited on pages 45, 52, and 55)

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. (Cited on pages 38, 51, 52, 55, 69, 82, 129, and 148)

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2012b). The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537–2548. (Cited on pages 19, 46, 52, 55, and 69)

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2013). Thirty years of inventory routing. *Transportation Science*, 48(1):1–19. (Cited on pages 20, 52)

Coelho, L. C. and Laporte, G. (2013). The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, 40(2):558–565. (Cited on pages 44, 45, 52, and 55)

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(Supplement C):391–397. (Cited on pages 20, 44, 52, and 55)

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91. (Cited on page 48)

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. (Cited on pages 20, 52, and 55)

Diabat, A., Bianchessi, N., and Archetti, C. (2024). On the zero-inventory-ordering policy in the inventory routing problem. *European Journal of Operational Research*, 312(3):1024–1038. (Cited on pages 35, 36, 52, and 55)

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271. (Cited on page 72)

Dinh, N. M., Archetti, C., and Bertazzi, L. (2023). The inventory routing problem with split deliveries. *Networks*, 82(4):400–413. (Cited on pages 46, 52, and 55)

Dror, M. and Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics (NRL)*, 37(3):383–402. (Cited on page 46)

Duhamel, C., Lacomme, P., and Prodhon, C. (2012). A hybrid evolutionary local search with depth first search split procedure for the heterogeneous vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 25(2):345–358. Special Section: Local Search Algorithms for Real-World Scheduling and Planning. (Cited on pages 50, 72)

Duhamel, C., Lacomme, P., Quilliot, A., and Toussaint, H. (2011). A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research*, 38(3):617–640. (Cited on pages 90, 138)

Eksioglu, B., Vural, A. V., and Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483. (Cited on page 38)

Farias, K., Hadj-Hamou, K., and Yugma, C. (2019). Mathematical formulations for a two-echelon inventory routing problem. *IFAC-PapersOnLine*, 52(13):1996–2001. (Cited on pages 47, 53, and 55)

Fisher, M. L. (1981). The lagrangian relaxation method for solving integer programming problems. *Management science*, 27(1):1–18. (Cited on page 24)

Gendreau, M., Laporte, G., Musaraganyi, C., and Taillard, É. D. (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 26(12):1153–1173. (Cited on page 38)

Glover, F. (1990). Tabu search: A tutorial. *Interfaces*, 20(4):74–94. (Cited on page 45)

Goel, A. and Gruhn, V. (2008). A general vehicle routing problem. *European Journal of Operational Research*, 191(3):650–660. (Cited on page 38)

Golden, B., Assad, A., Levy, L., and Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66. (Cited on page 38)

Guemri, O., Bekrar, A., Beldjilali, B., and Trentesaux, D. (2016). Grasp-based heuristic algorithm for the multi-product multi-vehicle inventory routing problem. *4OR-Q J Oper Res*, 14:377–404. (Cited on pages 53, 55, 90, and 138)

Guimarães, T. A., Coelho, L. C., Schenekemberg, C. M., and Scarpin, C. T. (2019). The two-echelon multi-depot inventory-routing problem. *Computers & Operations Research*, 101:220–233. (Cited on pages 47, 53, and 55)

Gutierrez-Alcoba, A., Rossi, R., Martin-Barragan, B., and Embley, T. (2023). The stochastic inventory routing problem on electric roads. *European Journal of Operational Research*, 310(1):156–167. (Cited on pages 20, 53, and 56)

Haddad, M. N., Martinelli, R., Vidal, T., Martins, S., Ochi, L. S., Souza, M. J. F., and Hartl, R. (2018). Large neighborhood-based metaheuristic and branch-and-price for the pickup and delivery problem with split loads. *European Journal of Operational Research*, 270(3):1014–1027. (Cited on page 44)

Hemmati, A., Hvattum, L. M., Christiansen, M., and Laporte, G. (2016). An iterative two-phase hybrid matheuristic for a multi-product short sea inventory-routing problem. *European Journal of Operational Research*, 252(3):775–788. (Cited on pages 44, 53, and 56)

Hoff, A., Andersson, H., Christiansen, M., Hasle, G., and Løkketangen, A. (2010). Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37(12):2041–2061. (Cited on page 38)

Koç, Ç., Bektaş, T., Jabali, O., and Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, 249(1):1–21. (Cited on page 39)

Kontoravdis, G. and Bard, J. F. (1995). A grasp for the vehicle routing problem with time windows. *ORSA Journal on Computing*, 7(1):10–23. (Cited on pages 90, 138)

Lacomme, P., Prins, C., and Ramdane-Chérif, W. (2001). A genetic algorithm for the capacitated arc routing problem and its extensions. In Boers, E. J. W., Gottlieb, J., Lanzi, P. L., Smith, R. E., Cagnoni, S., Hart, E., Raidl, G. R., and Tijink, H., editors, *Applications of Evolutionary Computing, EvoWorkshops 2001: EvoCOP, EvoFlight, EvoIASP, EvoLearn, and EvoSTIM, Como, Italy, April 18-20, 2001, Proceedings*, volume 2037 of *Lecture Notes in Computer Science*, pages 473–483. Springer. (Cited on page 21)

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. (Cited on pages 19, 46, 53, and 56)

Li, J., Chu, F., and Chen, H. (2011). Coordination of split deliveries in one-warehouse multi-retailer distribution systems. *Computers & Industrial Engineering*, 60(2):291–301. (Cited on page 46)

Manousakis, E., Repoussis, P., Zachariadis, E., and Tarantilis, C. (2021). Improved branch-and-cut for the inventory routing problem based on a two-commodity flow formulation. *European Journal of Operational Research*, 290(3):870–885. (Cited on pages 44, 53, 56, 83, and 121)

Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329. (Cited on page 67)

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. (Cited on pages 45, 53, and 56)

Molina, J. C., Salmeron, J. L., Eguia, I., and Racero, J. (2020). The heterogeneous vehicle routing problem with time windows and a limited number of resources. *Engineering Applications of Artificial Intelligence*, 94:103745. (Cited on page 38)

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. (Cited on pages 19, 46, 53, and 56)

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002. (Cited on pages 48, 92, and 111)

Prins, C., Lacomme, P., and Prodhon, C. (2014). Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C: Emerging Technologies*, 40:179–200. (Cited on pages 17, 47, 48, and 50)

Ramkumar, N., Subramanian, P., Narendran, T. T., and Ganesh, K. (2012). Mixed integer linear programming model for multi-commodity multi-depot inventory routing problem. *OPSEARCH*, 49:413–429. (Cited on page 19)

Rohmer, S., Claassen, G., and Laporte, G. (2019). A two-echelon inventory routing problem for perishable products. *Computers & Operations Research*, 107:156–172. (Cited on pages 45, 53, and 56)

Roldán, R. F., Basagoiti, R., and Coelho, L. C. (2017). A survey on the inventory-routing problem with stochastic lead times and demands. *Journal of Applied Logic*, 24(Part A):15–24. (Cited on page 20)

Schenekemberg, C. M., Scarpin, C. T., Pécora Jr, J. E., Guimarães, T. A., and Coelho, L. C. (2020). The two-echelon inventory-routing problem with fleet management. *Computers & Operations Research*, 121:104944. (Cited on pages 46, 53, and 56)

Shaabani, H. (2022). A literature review of the perishable inventory routing problem. *The Asian Journal of Shipping and Logistics*, 38(3):143–161. (Cited on page 45)

Skålnes, J., Ben Ahmed, M., Hvattum, L. M., and Stålhane, M. (2023). New benchmark instances for the inventory routing problem. *European Journal of Operational Research.* (Cited on pages 20, 38, 51, 54, 55, 56, 69, 120, 121, and 122)

Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., and van der Vorst, J. G. (2015). Modeling an inventory routing problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics*, 164:118–133. (Cited on pages 20, 54, and 56)

Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., and van der Vorst, J. G. (2018). Modeling a green inventory routing problem for perishable products with horizontal collaboration. *Computers & Operations Research*, 89:168–182. (Cited on pages 45, 54, and 56)

Soysal, M., Çimen, M., Belbağ, S., and Toğrul, E. (2019). A review on sustainable inventory routing. *Computers & Industrial Engineering*, 132:395–411. (Cited on page 20)

Toth, P. and Vigo, D. (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123(1-3):487–512. (Cited on page 38)

Touzout, F. A., Ladier, A.-L., and Hadj-Hamou, K. (2022). An assign-and-route matheuristic for the time-dependent inventory routing problem. *European Journal of Operational Research*, 300(3):1081–1097. (Cited on pages 44, 54, and 56)

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., and Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858. (Cited on page 38)

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., and Velasco, N. (2011). A grasp with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 38(9):1319–1334. (Cited on pages 90, 138)

Wilson, J. M. (1991). Supplier credit in the economic order quantity model. *International Journal of Operations & Production Management*, 11(9):64–71. (Cited on page 20)

Wolf, S. and Merz, P. (2007). Evolutionary local search for the super-peer selection problem and the p-hub median problem. In *International workshop on hybrid metaheuristics*, pages 1–15. Springer. (Cited on page 108)

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*, 197:135–158. (Cited on pages 46, 54, and 56)

Zhang, Z., Wei, L., and Lim, A. (2015). An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints. *Transportation Research Part B: Methodological*, 82:20–35. (Cited on pages 90, 138)